Fast MCMC algorithms, Stability and DeepTune

by

Yuansi Chen

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Statistics

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Bin Yu, Chair
Professor Martin J. Wainwright
Professor Jack L. Gallant

Summer 2019

Fast MCMC algorithms, Stability and DeepTune

Abstract

Fast MCMC algorithms, Stability and DeepTune

by

Yuansi Chen

Doctor of Philosophy in Statistics

University of California, Berkeley

Professor Bin Yu, Chair

Drawing samples from a known distribution is a core computational challenge common to many disciplines, with applications in statistics, probability, operations research, and other areas involving stochastic models. In statistics, sampling methods are useful for both estimation and inference, including problems such as estimating expectations of desired quantities, computing probabilities of rare events, gauging volumes of particular sets, exploring posterior distributions and obtaining credible intervals etc.

Facing massive high dimensional data, both computational efficiency and good statistical guarantees are more and more important in modern statistical and machine learning applications. In this thesis, centered around sampling algorithms, we consider the fundamental questions on their computational and statistical guarantees: How to design a fast sampling algorithm and how long should it be run? What are the statistical learning guarantee of these algorithms? Are there any trade-offs between computation and learning?

To answer these questions, first we start with establishing non-asymptotic convergence guarantees for popular MCMC sampling algorithms in Bayesian literature: Metropolized Random Walk, Metropolis-adjusted Langevin algorithm and Hamiltonian Monte Carlo. To address a number of technical challenges arise enroute, we develop results based on the conductance profile in order to prove quantitative convergence guarantees general continuous state space Markov chains. Second, to confront a large class of constrained sampling problems, we introduce two new algorithms, Vaidya and John walks, to sample from polytope-constrained distributions with convergence guarantees. Third, we prove fundamental trade-off results between statistical learning performance and convergence rate of any iterative learning algorithm, including sample algorithms. The trade-off results allow us to show that a too stable algorithm can not converge too fast, and vice-versa. Finally, to help neuroscientists analyze their massive amount of brain data, we develop DeepTune, a stability-driven visualization and interpretation framework via optimization and sampling for the neural-network-based models of neurons in visual cortex.

To my family

Contents

List of Figures

List of Tables

Acknowledgments

My six years of Ph.D. life at UC Berkeley would not be so vibrant without the presence of plentiful talented researchers and professionals. For this thesis, I owe a great amount of credit to these people who have supported, helped or inspired me.

First of all, I must thank my advisor Bin Yu. It has become more and more clear to me toward the end of my Ph.D. that choosing Bin as an advisor is the best choice I could have ever made in my life. Bin is a role model of mine in many facets. She is the perfect example of hard-working researchers in academia. She showed me how it is possible to manage a research group even when her physical condition was at stake. Not only is she a great researcher, but also is she a good trainer. Bin has been constantly pushing me into multidiscipline ressearch and reminding me to step out my comfort zone. Without her constant encouragement, I would have neither been in love with collaborative research in neuroscience nor learned so much from people in other fields. Without her backing, I would not have taken these chanllenging directions in my thesis. Her extensive data wisdom is extremely valuable. Every time I talk to her about real data problems, I feel lucky to steal a galance at her data wisdom. Teaching undergraduate maching learning together with her was also a great experiencet. It made me understand her philosophy and her strong sense of responsibility to connect staistics teaching with reality more closely. Imperceptibly, she has shaped my vision of the field. Looking back to my first years, I now realize that it must require tremendous patience and empathy to be able to train soneone as stubborn as me. It was the best experience in my life to have worked with her over this long period.

Martin Wainwright and Jack Gallant are my other academic mentors: these two professors who have the most incluence on me other than Bin. I first met Martin by taking his graduate class on theoretical statistics. I was immediately attracted because I had never met a professor who was able to cover profound theoretical materials in such a simple and concise fashion. Later when I started talking to him about research, I was further impressed by his deep theoretical understanding of many problems and his mathematical sharpness. While I learnt a lot skills for writing technical research papers from Martin, Jack is the mentor who guided me into broad-audience scientific research writing. Jack's passion for making scientific evidence and arguments easy to digest helped me develop a good taste of scientific wrting. Both Martin and Jack have provided great environments for communicating and discussing research: Martin's group meeting was full of new unprecedented inspirations; Jack's group meetings often dragged me out of my mathematical world and made me think harder what kind of statistics research is the most relevant for applied fields.

I am fortunate to have an advisor who has many academic and personal connections in our universities and research institutes. During my Ph.D., I had serveral chances to visit other research groups and observe closely different ways of doing research. I am indebted to Julien Mairal for getting me started in the research before my Ph.D. and eventually recommending me to pursue the study with Bin. I still look up to Julien's remarkable talent for rapidly distilling the essence of a theoretical problem.

My second research internship happens at Flatiron institute where I closely worked with Mitya Chklovskii. I appreciate Mitya's patience and support of my ideas. His extensive experience in the field allowed me to solidify my theoretical neuroscience understandings, which eventually attracted me into neuroscience reserach later with Jack at Berkeley. I would like to thank Peter Bühlmann for hosting me during my visit to ETH Zürich in my fourth year. Peter opened my eyes to a new research direction and also made my stay very enjoyable.

There are a number of other faculty at Berkeley from whom I learn a lot. I am very grateful for the group meetings of Martin Wainwright, Peter Bartlett, Jack Gallant and the causal reading group hosted by Peng Ding, Sam Pimental and Will Fithian. These meetings were always full of intense discussions. I enjoyed putting myself into a thinking position. I would like to thank Will Fithian and Aditya Guntobuyina for being on my qualifying exam committee and giving valuable feedback along the way. Teaching applied statistic course with Phillip Stark convinced me the importance of teaching students to differentiate good and bad applied statistics in our life. I am also thankful for the teaching experience with Jitendra Malik, for embracing together the transition of the computer vision teaching from classics to deep neural networks. Peter Bickel, who taught me a lot about semi-parametric estimation and provided great feedback on serveral of my ideas.

UC Berkeley is full talented graduate students. I feel fortunate to be surrounded by brilliant ideas. I must thank Raaz Dwivedi, who is not only a great colleague but also a close caring friend. Raaz Dwivedi's sharp mathematical thinking and excellent organization skills made it possible for us to quickly navigate the field of MCMC sampling, resulting in a number of publications. Raaz is also a fun person to work with: I enjoyed the laughters that he was always able to brought in during our often tiresome technical discussions. I am also grateful for Reza Abbasi for bearing with me my crazy ideas in our early neuroscience research. I appreciated the deep research discussions with him so much that we often forgot to the dinner time in the evening. Adam Bloniarz, as a great mentor who guided me into the neuroscience research. Chi Jin, who always impressed me with his fast mathematical insights. Yian Ma for his dedication to simple and elegant mathematical proofs.

Besides my collaborators, numerous people at Berkeley had made my life much less monotone. Thanks to Jamie Murdoch and Rebecca Barter for being wonderful office mates. From both, I learned a ton of writing and communication skills. They helped me feel not alone during my slow-pace research. Thanks to Siqi Wu and Hongwei Li for invaluable advice to survive my first years. Dominik Rothenhäusler for taking me to visit the beautiful mountains in Switzerland and for lengthy philosophical discussions about the future of the field. Sören Künzel and Simon Walter for organizing relaxing afterwork group events. Yu Wang and Xiao Li for fruitful discussions. Lihua Lei for for always bringing in fresh research ideas. Wenpin Tang for being my probability back-up who is capable of nailing down any probability related questions. Orhan Ocal, Yuting Wei for making the time waiting for meetings with Martin much more enjoyable. Chris Paciorek and Ryan Lovett, our department's adorable server masters for always being

able to make all kinds of bugs and failures dispear in less than a couple of hours.

Last but not least, I owe the most to my family. I thank my parents for their love, the incredible support and the tolerance of me following unconventional career path in my family history. I must thank my wonderful wife, Biyue Pan. I feel lucky to have met her back during my high school and to have decided together to get married at the begining of my Ph.D. study. Biyue constantly sacrifies her personal time to take care of my job interview travel schedulings, to adapt to my weird sleeping schedules during paper writing periods and to organize memorable moments with my friends. She taught me the true essence of a happy life. My Ph.D. life would be much less colorful without her companion and her smile.

Part I

Introduction and Background

Chapter 1

Introduction

Drawing samples from a known distribution is a core computational challenge common to many disciplines with applications in statistics, probability, operations research, and other areas involving stochastic models. For example, sampling methods in Bayesian statistics for exploring posterior distributions [31, 181], in simulation-based methods for reinforcement learning, and in image synthesis in computer vision, among other areas. Markov Chain Monte Carlo (MCMC) dates back to the seminal work of Metropolis et al. [134], and is the method of choice for drawing samples from high-dimensional distributions. The fact that it allows to sample from distributions with intractable normalization makes it easy to implement and to use.

Recent advancements of technologies in data gathering, especially in bioinformatics, neuroscience and medicine, have generated a huge volume of high dimensional data. While we hope that more data should allow better understanding of the underlying scientific problem, it becomes more and more challenging to identify the most efficient way to analyze the data. Data analysts often face the problem of make correct model and algorithm choices that balance statistical and computational performance. As a consequence, it is important to have a better understanding of both the statistical and computational guarantees of existing optimization and sampling algorithms.

This thesis has three thrusts: first, we develop a theoretically-motivated framework to understand various MCMC sampling algorithms. We wish not only to master existing MCMC sampling algorithm both from the computational and statistical angles, but also to know how to design new efficient algorithms. Secondly, we mainly focus on characterizing precisely convergence speed, stability and generalization of these algorithms. Thirdly, we demonstrate in a real-world data example how efficient optimization and sampling algorithms could lead to better understanding of our brain function. In particular, we develop a DeepTune modeling and visualization framework to discover the pattern selectivity in area V4. In the following subsections, we outline the core problems and some of the key ideas that will be developed in the remainder of this thesis.

1.1 MCMC sampling: with or without Metropolis-Hastings? with or without gradient?

Recent decades have witnessed great success of Markov Chain Monte Carlo (MCMC) algorithms in generating random samples; for instance, see the handbook [23] and references therein. In a broad sense, these methods are based on two steps. The first step is to construct a Markov chain whose stationary distribution is either equal to the target distribution or close to it in a suitable metric. Given this chain, the second step is to draw samples by simulating the chain for a certain number of steps.

Many algorithms have been proposed and studied for sampling from probability distributions with a density on a continuous state space. Two broad categories of these methods are *zeroth-order methods* and *first-order methods*. On one hand, a zeroth-order method is based on querying the density of the distribution (up to a proportionality constant) at a point in each iteration. By contrast, a first-order method makes use of additional gradient information about the density. A few popular examples of zeroth-order algorithms include Metropolized random walk (MRW) [132, 167], Ball Walk [119, 59, 118] and the Hit-and-run algorithm [12, 95, 115, 120, 123]. A number of first-order methods are based on the Langevin diffusion. Algorithms related to the Langevin diffusion include the Metropolis adjusted Langevin Algorithm (MALA) [166, 165, 18], the unadjusted Langevin algorithm (ULA) [152, 72, 166, 43], underdamped Langevin MCMC [39], Riemannian MALA [202], Proximal-MALA [155, 56], Metropolis adjusted Langevin truncated algorithm [166], Hamiltonian Monte carlo [145] and Projected ULA [25]. More details can be found in the survey [164].

An alternative way to divide these algorithms is based on whether a Metropolis-Hastings filter is applied. The algorithms without a Metropolis-Hastings filter is called *unadjusted*. Unadjusted methods include the unadjusted Langevin algorithm (ULA), underdamped Langevin MCMC, unadjusted Hamiltonian Monte Carlo. The adjusted algorithms work similarly, except that a Metropolis-Hastings filter is added at the end of each iteration. The Metropolis-Hastings step ensures that the algorithm always has the correct stationary distribution.

Given the large variety of sampling algorithms, a natural question arise: how shall we choose among first-order, zeroth-order, adjusted or unadjusted algorithms when facing a practical sampling problem? A more serious question follows after choosing an algorithm: how to choose hyperparameters such as step-size to make sure the algorithm is run efficiently? To answer these questions, it is necessary to develop a theoretical framework to compare these different sampling algorithms and different hyperparameter settings.

Previously, the attempt to develop such a theoretical framework dates back to Roberts and Tweedie [166]. They derived sufficient conditions for exponential convergence of the Langevin diffusion and its discretizations, with and without Metropolis-adjustment. However, the distributions they consider are limited to k-th order mo-

ments. Since then, various convergence results have been established (e.g. [167, 18, 43, 55, 38]) with usually a focus on a single algorithm. With this context, our main goal in Chapter 2 is to provide an explicit convergence comparison for first-order vs. zeroth-order methods and adjusted vs. unadjusted methods. This chapter is based on joint work with Raaz Dwivedi, Martin Wainwright and Bin Yu [58].

1.2 Beyond simple MCMC algorithms

In order to handle more data and to deal with more complex and structured data, people are in constant need of fast algorithms that take advantage of the structure of the problem. In general, there are two types of structures for sampling algorithms: one that an algorithm can take advantage of to run faster than naive algorithms; the other that requires specific treatment that a naive algorithm can not even generate good enough samples.

In particular, if the target distribution one wants to sample from enjoys some high-order smoothness, can one design a faster algorithm than MRW or MALA? The answer is affirmative. It was first observed in chemical physics literature by Alder and Wainwright [3] that algorithms using Hamiltonian dynamics can converge much faster. The algorithm was refined by Neal [144], and later re-christened in statistics community as Hamiltonian Monte Carlo. While HMC enjoys fast convergence in practice, a theoretical understanding of this behavior remains incomplete. Some intuitive explanations are based on its ability to maintain a constant asymptotic accept-reject rate with large step-size (e.g. [42]). Others (e.g. Neal [145]) suggest, based on intuition from the continuous-time limit of the Hamiltonian dynamics, that HMC is able to suppress random walk behavior using momentum. However, these intuitive arguments do not provide rigorous or quantitative justification for the fast convergence of the discrete-time HMC used in practice. Our goal in Chapter 3 is to provide a non-asymptotic convergence guarantees for HMC algorithm so that one can tell precisely when and why it converges faster than simple algorithms such as MRW and MALA.

On the other hand, many applications require sampling from a distribution that is only defined on a constrained set. Naive sampling algorithms combined with rejection sampling will have high rejection rates in high dimension. How to efficiently tackle these constraints, such as polytope or convex body constraints, attracted a long line of work [119, 103, 12, 115, 40]. For polytope constraints, many MCMC algorithms have been studied. Some early examples include the Ball Walk [119] and the hit-and-run algorithm [12, 115], which apply to sampling from general convex bodies. Although these algorithms can be applied to polytopes, they do not exploit any special structure of the problem. In contrast, the Dikin walk introduced by [97] is specialized to polytopes, and thus can achieve faster convergence rates than generic algorithms. The Dikin walk was the first sampling algorithm based on a connection to interior point methods for solving linear programs. More specifically, as we discuss in detail below, it constructs proposal distributions based on the standard logarithmic barrier for a polytope. One

main drawback of Dikin walk is that its convergence scales linearly as the number of linear constraints in a polytope increases. Is it possible to design a sampling algorithm whose mixing time scales in a sub-linear manner with the number of constraints? Our main goal in Chapter 4 is to investigate and answer this question in affirmative—in particular, by designing and analyzing two sampling algorithms with provably faster convergence rates than the the Dikin walk while retaining its advantages over the ball walk and the hit-and-run methods. Both Chapter 3 and Chapter 4 are based on joint work with Raaz Dwivedi, Martin Wainwright and Bin Yu [58, 36, 35].

1.3 Stability considerations

In statistics and machine learning, the computational concerns are important only when good statistical performance is first met. For different supervised learning algorithms ranging from classical linear regression, logistic regression, boosting, to modern large-scale deep networks, the overall performance or expected excess risk can always be decomposed into two parts: the empirical error (or the training error) and the generalization error (characterizing the discrepancy between the test error and the training error). A central theme in statistics and machine learning is to find an appropriate balance between empirical error and generalization error, because improperly emphasizing one over the other typically results in either overfitting or underfitting.

Traditionally, these two quantities are mostly studied separately. On one hand, the empirical error is controlled by convergence analysis in optimization and sampling theory. Recent years have witnessed a rapid advance on convergence rates analysis of specific optimization methods for a particular class of loss functions that they are optimizing over. In fact, such analysis has been carried out for many gradient methods, including gradient descent (GD), Nesterov accelerated gradient descent (NAG), stochastic gradient descent (SGD), stochastic gradient Langevin dynamics (SGLD) for convex, strongly convex, or even non-convex functions (see e.g. [21, 24, 149, 91, 158]).

On the other hand, the generalization error can be handled by algorithmic stability analysis. Algorithmic stability [50, 20] in learning problems has been introduced as an alternative way to control generalization error instead of uniform convergence results such as classical VC-theory [192] and Rademacher complexity [11]. The stability concept has an intuitive appeal: an algorithm is stable if it is robust to small perturbations in the composition of the learning data set. Recently it has been shown that algorithmic stability is well suited for controlling generalization error of stochastic gradient methods [75], as well as stochastic gradient Langevin dynamics algorithm [138].

In a specific statistical problem, unless the optimization error and generalization error of these algorithms are analyzed together, it is not clear whether the fastest converging optimization algorithm is the best for learning. Our goal in Chapter 5 is to characterize the trade-off between the convergence rate and the algorithmic stability of iterative algorithms. This chapter is based on joint work with Chi Jin and Bin Yu [34].

1.4 Efficient computation to interpret neuron models

Understanding how primates process visual information and recognize objects in an image is a major problem in neuroscience. Along the visual pathway, the mid-tier cortical area V4 is of particular interest. Despite its importance in the hierarchical organization of visual processing, its function remains elusive. Deep neural network models have recently been shown to be effective in predicting single neuron responses in primate visual cortex areas V4 [204, 28, 203]. While this deep, convolutional and non-linear architecture is the key to the high predictive performance, it also makes the models difficult to interpret. This limits their usefulness in advancing neuroscience.

A natural question arises: can we use these complex and accurate models to infer tuning properties of V4 neurons? Our goal in Chapter 6 is to develop efficient optimization or sampling methods to visualize and interpret these colossal neural-network-based predictive models. We propose the DeepTune framework to interpret deep neural network-based models of single neurons in area V4. Using a dataset of recordings of 71 V4 neurons stimulated with thousands of static natural images, we first build an ensemble of 18 neural network-based models per neuron accurately predict its response given a stimulus image. These models achieve the state-of-the-art prediction performance. To leverage the good performance to understand V4 neurons better, we use a stability criterion to form optimal stimuli (DeepTune images) by pooling the 18 models together. These DeepTune images not only provide concrete visualization of shape and texture tuning in area V4, but also create naturalistic stimuli for future closed-loop experiments. This chapter is based on joint work with Reza Abbasi-Asl, Adam Bloniarz, Michael Oliver, Ben D.B. Willmore, Jack L. Gallant and Bin Yu [1].

Organization: The rest of the thesis is organized as follows: Part I of this thesis is concerned with computational guarantees of MCMC sampling algorithms. We start by setting up the MCMC sampling problem and providing background on proof techniques in Chapter 2. The technical notations and tools provided are relevant for the entire Part I. Chapter 3 goes beyond simple sampling algorithms, and consider the state-of-the-art Hamiltonian Monte Carlo algorithm. Chapter 4 takes a different direction and considers sampling from distributions constrained on polytopes.

Part II of this thesis focus on statistical aspect of iterative algorithms. In Chapter 5, we investigate the stability guarantees in addition to computation guarantees and the trade-off between these two quantities.

Part III of this thesis study the functionality of neurons in visual cortex area V4. This is an inter-discipline collaborative effort between neuroscience and statistics to advance the understanding of our brain. Chapter 6 describes how we build our convolutional neural network based models, how we interpret them using efficient gradient-based optimization and sampling algorithms and why the DeepTune images reveal fundamental properties of V4 encoding.

1.5 Notations

Here we define notation and terminology that we commonly use.

We use $\mathbb{R}$ to denote the set of real numbers and $\mathbb{N}$ to denote the set of natural numbers. We use $[K]$ to denote the integers from the set $\{1, 2, \ldots, K\}$. d is used as dimension unless otherwise stated. For two real-valued sequences $\{a_n\}_{n\in\mathbb{N}}$ and $\{b_n\}_{n\in\mathbb{N}}$, we write $a_n = O(b_n)$ if there exists a universal constant $c > 0$ such that $a_n \leq cb_n$. We write $a_n = \widetilde{O}(b_n)$ if $a_n \leq c_n b_n$, where c_n grows at most poly-logarithmically in n. Throughout we use the notation c, c_1, c_2 to denote universal constants.

For a vector $x \in \mathbb{R}^d$, we use $\|x\|_{\mathfrak{p}}$ to denote its $\ell_{\mathfrak{p}}$-norm $\|x\|_{\mathfrak{p}} = \left(\sum_{j=1}^{d} |x_j|^{\mathfrak{p}}\right)^{\frac{1}{\mathfrak{p}}}$. For a matrix $A \in \mathbb{R}^{d_1\times d_2}$, we use $|\!|\!|A|\!|\!|_2$ to denote its maximum singular value, i.e.

$$|\!|\!|A|\!|\!|_2 := \sup_{v\in\mathbb{R}^{d_2}, \|v\|_2\leq 1} \|Av\|_2 .$$

Probability: We use $\mathcal{X}$ to denote the (general) state space of a Markov chain. We denote $\mathcal{B}(\mathcal{X})$ as the Borel σ-algebra of the state space $\mathcal{X}$. Given two probability distribution P and Q on the state space $\mathcal{X}$, assumed absolutely continuous with respect to the Borel measure with densities p and q, the KL-divergence between P and Q is defined as

$$\mathrm{KL}(P \parallel Q) = \int_{\mathcal{X}} p(x) \log \frac{p(x)}{q(x)} dx.$$

We define the $\mathcal{L}_{\mathfrak{p}}$-divergence ($\mathfrak{p} \geq 1$) of P with respect to the distribution Q as

$$d_{\mathfrak{p}}(P, Q) = \left(\int_{\mathcal{X}} \left| \frac{p(x)}{q(x)} - 1 \right|^{\mathfrak{p}} q(x) dx \right)^{\frac{1}{\mathfrak{p}}}. \tag{1.1}$$

For $\mathfrak{p} = 2$, we get the χ^2-divergence. For $\mathfrak{p} = 1$, the distance $d_1(P, Q)$ represents two times the total variation distance between P and Q. In order to make this distinction clear, we use $d_{\mathrm{TV}}(P, Q)$ to denote the total variation distance.

We use $\mathcal{N}(\mu, \Sigma)$ to denote the normal distribution with mean μ and covariance matrix Σ.

Derivatives: For a three-times differentiable function $f : \mathbb{R}^d \to \mathbb{R}$, we represent its first, second, third derivatives at $x \in \mathbb{R}^d$ by $\nabla f(x) \in \mathbb{R}^d$, $\nabla^2 f(x) \in \mathbb{R}^{d\times d}$ and $\nabla^3 f(x) \in \mathbb{R}^{d^3}$. Here

$$[\nabla f(x)]_i = \frac{\partial}{\partial x_i} f(x), \quad \left[\nabla^2 f(x)\right]_{i,j} = \frac{\partial^2}{\partial x_i \partial x_j} f(x), \quad \left[\nabla^3 f\right]_{i,j,k} = \frac{\partial^3}{\partial x_i \partial x_j \partial x_k} f(x).$$

A function $f : \mathbb{R}^d \to \mathbb{R}$ is said to be M-Lipschitz continuous if

$$|f(x) - f(y)| \leq M \|x - y\|_2 \quad \text{for all } x, y \in \mathbb{R}^d. \tag{1.2a}$$

Similarly, a differentiable f is said to be L-smooth if its gradient ∇f is L-Lipschitz continuous, i.e.

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L \|x - y\|_2 \quad \text{for all } x, y \in \mathbb{R}^d. \tag{1.2b}$$

A twice-differentiable f is said to be L_{H}-Hessian Lipschitz if

$$\left\|\left|\nabla^2 f(x) - \nabla^2 f(y)\right|\right\|_2 \leq L_{\mathrm{H}} \|x - y\|_2 \quad \text{for all } x, y \in \mathbb{R}^d. \tag{1.2c}$$

A set $\Omega \subset \mathbb{R}^d$ is convex if $x, y \in \Omega$ implies $\lambda x + (1 - \lambda) y \in \Omega$ for all $\lambda \in [0, 1]$. A function f is convex if

$$f(\lambda x + (1 - \lambda) y) \leq \lambda f(x) + (1 - \lambda) f(y) \quad \text{for all } x, y \in \mathbb{R}^d. \tag{1.2d}$$

Furthermore, convex function f is said to be m-strongly convex if

$$f(y) - f(x) - \nabla f(x)^\top (y - x) \geq \frac{m}{2} \|x - y\|_2^2 \qquad \text{for all } x, y \in \mathbb{R}^d. \tag{1.2e}$$

Part II

Computational aspects of sampling

Chapter 2

Random walk and Langevin algorithms

In this part of the thesis, we focus on the problem of drawing samples from a distribution over an unconstrained continuous state space. We consider a distribution Π^* defined over $\mathcal{X}$ with density $\pi^* : \mathcal{X} \to \mathbb{R}_+$, specified explicitly up to a normalization constant as follows

$$\pi^*(x) \propto e^{-f(x)}. \tag{2.1}$$

The sampling problem arises typically when we want to estimate the expectation of some function $g : \mathcal{X} \to \mathbb{R}$ – that is, to approximate

$$\Pi^*(g) = \mathbb{E}_{\pi^*}[g(X)] = \int_{\mathcal{X}} g(x)\pi^*(x)dx. \tag{2.2}$$

For example, g can be linear or quadratic function if we want to estimate the mean or variance of the posterior distribution in Bayesian inference. The problem (2.2) is challenging in general: analytical computation of the integral (2.2) is infeasible; in high dimensional space, numerical integration is not feasible either due to the well-known curse of dimensionality.

A Monte Carlo approximation to $\Pi^*(g)$ is based on access to a sampling algorithm that can generate i.i.d. random variables $Z_i \sim \pi^*$ for $i = 1, \ldots, N$. Given such samples, the random variable $\widehat{\Pi^*}(g) := \frac{1}{N}\sum_{i=1}^{N} g(Z_i)$ is an unbiased estimate of the quantity $\Pi^*(g)$, and has its variance proportional to $1/N$. The challenge of implementing such a method is drawing the i.i.d. samples Z_i. If π^* has a complicated form and the dimension d is large, it is difficult to generate i.i.d. samples from π^*. For example, rejection sampling [70], which works well in low dimensions, fails due to the curse of dimensionality.

The Markov chain Monte Carlo (MCMC) approach is to construct a Markov chain on $\mathcal{X}$ that starts from some easy-to-simulate initial distribution μ_0, and converges to π^* as its stationary distribution. Two natural questions arise for the Markov chain construction:

1. how to design such chains?
2. how many steps will the Markov chain take to converge close enough to the stationary distribution?

Over the years, these questions have been the subject of considerable research; for instance, see the reviews [188, 180, 164] and references therein. In the coming chapters of this thesis, we illustrate a few answers to these questions by focusing on three popular Metropolis-Hastings adjusted Markov chains sampling algorithms: Metropolized random walk (MRW), Metropolis-adjusted Langevin algorithm (MALA) and Hamiltonian Monte Carlo (HMC).

In this chapter, we are particularly interested in establishing convergence rates of the gradient-free algorithm MRW and the gradient based Metropolized algorithm MALA for sampling from log-concave distributions. Log-concave distribution is a rich class of distributions. Standard examples of log-concave distributions include the normal distribution, exponential distribution and Laplace distribution. Comparing the convergence rates for sampling from log-concave distributions allow us to understand the benefits of gradient information and the Metropolis-Hastings filters in sampling algorithm design.

2.1 Introduction

Drawing samples from a known distribution is a core computational challenge common to many disciplines, with applications in statistics, probability, operations research, and other areas involving stochastic models. In statistics, these methods are useful for both estimation and inference. Under the frequentist inference framework, samples drawn from a suitable distribution can form confidence intervals for a point estimate, such as those obtained by maximum likelihood. Sampling procedures are also standard in the Bayesian setting, used for exploring posterior distributions, obtaining credible intervals, and solving inverse problems. Estimating the mean, posterior mean in a Bayesian setting, expectations of desired quantities, probabilities of rare events and volumes of particular sets are settings in which Monte Carlo estimates are commonly used.

Recent decades have witnessed great success of Markov Chain Monte Carlo (MCMC) algorithms in generating random samples; for instance, see the handbook [23] and references therein. In a broad sense, these methods are based on two steps. The first step is to construct a Markov chain whose stationary distribution is either equal to the target distribution or close to it in a suitable metric. Given this chain, the second step is to draw samples by simulating the chain for a certain number of steps.

Many algorithms have been proposed and studied for sampling from probability distributions with a density on a continuous state space. Two broad categories of these methods are *zeroth-order methods* and *first-order methods*. On one hand, a zeroth-order method is based on querying the density of the distribution (up to a proportionality constant) at a point in each iteration. By contrast, a first-order method makes

use of additional gradient information about the density. A few popular examples of zeroth-order algorithms include Metropolized random walk (MRW) [132, 167], Ball Walk [119, 59, 118] and the Hit-and-run algorithm [12, 95, 115, 120, 123]. A number of first-order methods are based on the Langevin diffusion. Algorithms related to the Langevin diffusion include the Metropolis adjusted Langevin Algorithm (MALA) [166, 165, 18], the unadjusted Langevin algorithm (ULA) [152, 72, 166, 43], underdamped Langevin MCMC [39], Riemannian MALA [202], Proximal-MALA [155, 56], Metropolis adjusted Langevin truncated algorithm [166], Hamiltonian Monte carlo [145] and Projected ULA [25]. There is now a rich body of work on these methods. More details can be found in the survey [164], which covers MCMC algorithms for general distributions, and the survey [193], which focuses on random walks for compactly supported distributions.

Here we focus on sampling algorithms for sampling from a log-concave distribution Π^* with density of the form

$$\Pi^*(x) = \frac{e^{-f(x)}}{\int_{\mathbb{R}^d} e^{-f(y)} dy} \quad \text{for all } x \in \mathbb{R}^d, \tag{2.3}$$

where f is a convex function on $\mathbb{R}^d$. Some recent work have provided non-asymptotic bounds on the mixing times of Langevin type algorithms for sampling from a log-concave density. The mixing time corresponds to the number of steps, as function of both the problem dimension d and the error tolerance ϵ, to obtain a sample from a distribution that is ϵ-close to the target distribution in total variation distance or other distribution distances. It is known that both the ULA updates [43, 55, 38] as well as underdamped Langevin MCMC [39] have mixing times that scale polynomially in the dimension d, as well the inverse of the error tolerance $1/\epsilon$.

Both the ULA and underdamped-Langevin MCMC methods are based on evaluations of the gradient ∇f, along with the addition of Gaussian noise. Durmus and Moulines [55] show that for an appropriate decaying step size schedule, the ULA algorithm converges to the right stationary distribution. However, their results, albeit non-asymptotic, are hard to quantify. In the sequel, we limit our discussion to Langevin algorithms based on constant step sizes, for which there are a number of explicit quantitative bounds on the mixing time. When one uses a fixed step size for these algorithms, an important issue is that the resulting random walks are asymptotically biased: due to the lack of Metropolis-Hastings correction step, the algorithms *will not* converge to the stationary distribution if run for a large number of steps. Furthermore, if the step size is not chosen carefully the chains may become transient [166]. Thus, typical theory is based on running such a chain for a pre-specified number of steps, depending on the tolerance, dimension and other problem parameters.

In contrast, the Metropolis-Hastings step that underlies the MALA algorithm ensures that the resulting random walk has the correct stationary distribution. Roberts and Tweedie [166] derived sufficient conditions for exponential convergence of the

Langevin diffusion and its discretizations, with and without Metropolis-adjustment. However, they considered the distributions with $f(x) = \|x\|_2^\alpha$ and proved geometric convergence of ULA and MALA under some specific conditions. In a more general setting, Bou-Rabee and Hairer [18] derived non-asymptotic mixing time bounds for MALA. However, all these bounds are non-explicit, and so makes it difficult to extract an explicit dependence in terms of the dimension d and error tolerance ϵ. A precise characterization of this dependence is needed if one wants to make quantitative comparisons with other algorithms, including ULA and other Langevin-type schemes. Along this note, Eberle [60] derived mixing time bounds for MALA albeit in a more restricted setting compared to the one considered in this work. In particular, Eberle's convergence guarantees are in terms of a modified Wasserstein distance, truncated so as to be upper bounded by a constant, for a subset of strongly concave measures which are four-times continuously differentiable and satisfy certain bounds on the derivatives up to order four. With this context, one of the main contributions of our work is to provide an explicit upper bound on the mixing time bounds in total variation distance of the MALA algorithm for general log-concave distributions.

Our contributions: This chapter contains two main results, both having to do with the mixing times of MCMC methods for sampling. As described above, our first and primary contribution is an explicit analysis of the mixing time of Metropolis adjusted Langevin Algorithm (MALA). A second contribution is to use similar techniques to analyze a zeroth-order method called Metropolized random walk (MRW) and derive a explicit non-asymptotic mixing time bound for it. Unlike the ULA, these methods make use of the Metropolis-hastings accept-reject step and consequently converge to the target distributions in the limit of infinite steps. Here we provide explicit non-asymptotic mixing time bounds for MALA and MRW and show that MALA converges significantly faster than ULA. In particular, we show that if the density is strongly log-concave and smooth, the ϵ-mixing time for MALA scales as $\kappa d \log(1/\epsilon)$ which is significantly faster than ULA's convergence rate of order $\kappa^2 d/\epsilon^2$. On the other hand, Moreover, we also show that MRW mixes $O(\kappa)$ slowly when compared to MALA. Furthermore, if the density is weakly log-concave, we show that (a modified version of) MALA converges in $O(d^2/\epsilon^{1.5})$ time in comparison to the $O(d^3/\epsilon^4)$ mixing time for ULA. As alluded to earlier, such a speed-up for MALA is possible since we can choose a large step size for it which in turn is possible due to its unbiasedness in the limit of infinite steps. In contrast, for ULA the step-size has to be small enough to control the bias of the distribution of the ULA iterates in the limit of infinite steps, leading to a relative slow down when compared to MALA.

2.2 Background and problem set-up

2.2.1 Markov chain basics

Let us now set up some basic notation and definitions on Markov chains that we use in the sequel. We consider *time-homogeneous* Markov chains defined on a measurable state space $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$ with a transition kernel $\Theta : \mathcal{X} \times \mathcal{B}(\mathcal{X}) \to \mathbb{R}_+$. By definition, the transition kernel satisfies the following properties:

$$\Theta(x, dy) \geq 0, \quad \text{for all } x \in \mathcal{X}, \quad \text{and} \quad \int_{y \in \mathcal{X}} \Theta(x, dy) dy = 1 \quad \text{for all } x \in \mathcal{X}.$$

The k-step transition kernel Θ^k is defined recursively

$$\Theta^{k+1}(x, dy) = \int_{z \in \mathcal{X}} \Theta^k(x, dz)\Theta(z, dy)dz.$$

The Markov chain is *irreducible* means that for all $x, y \in \mathcal{X}$, there is a natural number $k > 0$ such that $\Theta^k(x, dy) > 0$. We say that a Markov chain satisfies the *detailed balance condition* if

$$\pi^*(x)\Theta(x, dy)dx = \pi^*(y)\Theta(y, dx)dy \quad \text{for all } x, y \in \mathcal{X}. \tag{2.4}$$

Such a Markov chain is also called *reversible*. Finally, we say that a probability measure Π^* with density π^* on $\mathcal{X}$ is *stationary* (or *invariant*) for a Markov chain with the transition kernel Θ if

$$\int_{x \in \mathcal{X}} \pi^*(x)\Theta(y, dx) = \pi^*(y) \quad \text{for all } y \in \mathcal{X}.$$

Transition operator: We use $\mathcal{T}$ to denote the transition operator of the Markov chain on the space of probability measures with state space $\mathcal{X}$. In simple words, given a distribution μ_0 on the current state of the Markov chain, $\mathcal{T}(\mu_0)$ denotes the distribution of the next state of the chain. Mathematically, we have $\mathcal{T}(\mu_0)(A) = \int_{\mathcal{X}} \Theta(x, A)\mu_0(x)dx$ for any $A \in \mathcal{B}(\mathcal{X})$. In an analogous fashion, $\mathcal{T}^k$ stands for the k-step transition operator. We use $\mathcal{T}_x$ as the shorthand for $\mathcal{T}(\delta_x)$, the *transition distribution at* x; here δ_x denotes the Dirac delta distribution at $x \in \mathcal{X}$. Note that by definition $\mathcal{T}_x = \Theta(x, \cdot)$.

Mixing time of a Markov chain: Consider a Markov chain with initial distribution μ_0, transition operator $\mathcal{T}$ and a target distribution Π^* with density π^*. Its $\mathcal{L}_\mathfrak{p}$ mixing time with respect to Π^* is defined as follows:

$$\tau_\mathfrak{p}(\epsilon; \mu_0) = \inf \left\{ k \in \mathbb{N} \mid d_\mathfrak{p}\left(\mathcal{T}^k(\mu_0), \Pi^*\right) \leq \epsilon \right\}. \tag{2.5a}$$

where $\epsilon > 0$ is an error tolerance. Since distance $d_\mathfrak{p}(Q, \Pi^*)$ increases as $\mathfrak{p}$ increases, we have

$$\tau_\mathfrak{p}(\epsilon; \mu_0) \leq \tau_{\mathfrak{p}'}(\epsilon; \mu_0) \quad \text{for any} \quad \mathfrak{p}' \geq \mathfrak{p} \geq 1. \tag{2.5b}$$

Warm initial distribution: We say that a Markov chain with state space $\mathcal{X}$ and stationary distribution Π^* has a ϖ-*warm start* if its initial distribution μ_0 satisfies

$$\sup_{S \in \mathcal{B}(\mathcal{X})} \frac{\mu_0(S)}{\Pi^*(S)} \leq \varpi, \tag{2.6}$$

where $\mathcal{B}(\mathcal{X})$ denotes the Borel σ-algebra of the state space $\mathcal{X}$. For simplicity, we say that μ_0 is a warm start if the warmness parameter ϖ is a small constant (e.g., ϖ does not scale with dimension d).

Lazy chain: We say that the Markov chain is ζ-*lazy* if at each iteration the chain is forced to stay at the previous iterate with probability ζ. We study $\frac{1}{2}$-lazy chains in this chapter. In practice, one is not likely to use a lazy chain (since the lazy steps slow down the convergence rate by a constant factor); rather, it is a convenient assumption for theoretical analysis of the mixing rate up to constant factors.[1]

Metropolis-Hastings adjustment: We now briefly describe a certain class of Markov chains that are of *Metropolis-Hastings type* [134, 77]; see the books [161, 23] and references therein for further background.

Starting at a given initial density μ_0 over $\mathcal{X}$, any such Markov chain is simulated in two steps: (1) proposal step, and (2) accept-reject step. For the proposal step, we make use of a *proposal function* $\mathcal{P} : \mathcal{X} \times \mathcal{B}(\mathcal{X}) \to \mathbb{R}_+$, where $\mathcal{P}(x, \cdot)$ is a distribution for each $x \in \mathcal{X}$. At each iteration, given a current state $x \in \mathcal{X}$ of the chain, the algorithm proposes a new vector $z \in \mathcal{X}$ by sampling from the proposal distribution $\mathcal{P}(x, \cdot)$. In the second step, the algorithm accepts $z \in \mathcal{X}$ as the new state of the Markov chain with probability

$$\alpha(x, z) := \min \left\{ 1, \ \frac{\Pi^*(z)\mathcal{P}(z, dx)dz}{\Pi^*(x)\mathcal{P}(x, dz)dx} \right\}. \tag{2.7}$$

Otherwise, with probability equal to $1 - \alpha(x, z)$, the chain stays at x. Consequently, the overall transition kernel Θ for the Markov chain is defined by the function

$$\Theta(x, dz) := \mathcal{P}(x, dz)\alpha(x, z) \quad \text{for } z \neq x,$$

and a probability mass at x with weight $1 - \int_{\mathcal{X}} \Theta(x, dz)$. The purpose of the Metropolis-Hastings correction (2.7) is to ensure that the target density Π^* is stationary for the Markov chain.

[1] Any lazy (time-reversible) chain is always aperiodic and admits a unique stationary distribution. For more details, see the survey [193] and references therein.

2.2.2 From MRW to MALA

Given the set-up in the previous subsection, we now describe several algorithms for sampling from log concave distributions. Let $\mathcal{P}_x$ denote the proposal distribution at x corresponding to the proposal density $\mathcal{P}(x, \cdot)$. Possible choices of this proposal function include:

- Independence sampler: the proposal distribution does not depend on the current state of the chain, e.g., rejection sampling or when $\mathcal{P}_x = \mathcal{N}(0, \Sigma)$, where Σ is a hyper-parameter.
- Random walk: the proposal function satisfies $\mathcal{P}(x, dy) = \mathcal{P}(y, dx)$, e.g., when $\mathcal{P}_x = \mathcal{N}(x, 2\eta\mathbb{I}_d)$ where η is a hyper-parameter.
- Langevin algorithm: the proposal distribution is shaped according to the target distribution and is given by $\mathcal{P}_x = \mathcal{N}(x - \eta\nabla f(x), 2\eta\mathbb{I}_d)$, where η is chosen suitably.
- Symmetric Metropolis algorithm: the proposal function $\mathcal{P}$ satisfies $\mathcal{P}(x, dy) = \mathcal{P}(y, dx)$. Some examples are Ball Walk [63], and Hit-and-run [115].

Naturally the convergence rate of these algorithms would depend on the properties of Π^* and how well suited are the proposal distribution $\mathcal{P}$ for the task at hand. A key difference between Langevin algorithm and other algorithms is that the former makes use of the additional first order gradient information about the target distribution Π^*. We now briefly discuss the existing theoretical results about the convergence rate of different MCMC algorithms. Several results on MCMC algorithms have focused on on establishing behavior and convergence of these sampling algorithms in an asymptotic or a non-explicit sense, e.g., geometric and uniform ergodicity, asymptotic variance and central limit theorems. See the papers [184, 136, 167, 166, 88, 163, 165, 156, 162], the survey [164] and the references therein. Such results, albeit helpful for gaining insight, do not provide user-friendly rates of convergence. Consequently, from these results, it is not easy to determine the computational complexity of various MCMC algorithms as a function of the problem dimension d and desired accuracy ϵ. Explicit non-asymptotic convergence bounds, which provide useful information for practice, are the focus of this work.

Metropolized random walk

Metropolized random walk is based on Gaussian proposals. That is, when the chain is at state x_k, a proposal is drawn as follows

$$z_{k+1} = x_k + \sqrt{2\eta}\,\xi_{k+1}, \tag{2.8}$$

where the noise term $\xi_{k+1} \sim \mathcal{N}(0, \mathbb{I}_d)$ is independent of all past iterates. The chain then makes the transition according to an accept-reject step with respect to Π^*. Since

the proposal distribution is symmetric, this step can be described as

$$x_{k+1} = \begin{cases} z_{k+1} & \text{with probability } \min\left\{1, \dfrac{\Pi^*(z_{k+1})}{\Pi^*(x_k)}\right\} \\ x_k & \text{otherwise.} \end{cases}$$

This sampling algorithm is an instance of a zeroth-order method, since it makes use of only the function values of the density Π^*. We refer to this algorithm as MRW in the sequel. It is easy to see that the chain has positive density of jumping from any state x to y in $\mathbb{R}^d$ and hence is strongly Π^*-irreducible and aperiodic. Consequently, Theorem 1 by Diaconis et al. [51] implies that the chain has a unique stationary distribution Π^* and converges to in the limit of infinite steps. Note that this algorithms has also been referred to as Random walk Metropolized (RWM) and Random walk Metropolis-Hastings (RWMH) in the literature.

Langevin diffusion and related sampling algorithms

Langevin-type algorithms are based on Langevin diffusion, a stochastic process whose evolution is characterized by the stochastic differential equation (SDE):

$$dX_t = -\nabla f(X_t) + \sqrt{2}\, dW_t, \tag{2.9}$$

where $\{W_t \mid t \geq 0\}$ is the standard Brownian motion on $\mathbb{R}^d$. Under fairly mild conditions on f, it is known that the diffusion (2.9) has a unique strong solution $\{X_t, t \geq 0\}$ that is a Markov process [166, 135]. Furthermore, it can be shown that the distribution of X_t converges as $t \to +\infty$ to the invariant distribution Π^* characterized by the density $\Pi^* \propto \exp(-f)$. See Roberts and Tweedie [166] or Meyn and Tweedie [135] for further details.

Algorithm 1: Metropolized Random Walk (MRW)

Input: Step size $\eta > 0$ and a sample x_0 from a starting distribution μ_0
Output: Sequence $x_1, x_2, \ldots$

for $i = 0, 1, \ldots$ **do**
  Proposal step: *Draw* $z_{i+1} \sim \mathcal{N}(x_i, 2\eta \mathbb{I}_d)$
  Accept-reject step:
    compute $\alpha_{i+1} \leftarrow \min\left\{1, \dfrac{\exp\left(-f(z_{i+1}) - \|x_i - z_{i+1}\|_2^2 / 4\eta\right)}{\exp\left(-f(x_i) - \|z_{i+1} - x_i\|_2^2 / 4\eta\right)}\right\}$
    With probability α_{i+1} *accept* the proposal: $x_{i+1} \leftarrow z_{i+1}$
    With probability $1 - \alpha_{i+1}$ *reject* the proposal: $x_{i+1} \leftarrow x_i$
end

Unadjusted Langevin algorithm A natural way to simulate the Langevin diffusion (2.9) is to consider its forward Euler discretization, given by

$$x_{k+1} = x_k - \eta \nabla f(x_k) + \sqrt{2\eta}\xi_{k+1}, \tag{2.10}$$

where the driving noise $\xi_{k+1} \sim \mathcal{N}(0, \mathbb{I}_d)$ is drawn independently at each time step. The use of iterates defined by equation (2.10) can be traced back at least to Parisi in 1981 [152] for computing correlations as noted by Besag in his commentary on the paper by Grenander and Miller [72].

However, even when the SDE is well behaved, the iterates defined by this discretization have mixed behavior. For sufficiently small step sizes η, the distribution of the iterates defined by equation (2.10) converges to a stationary distribution that is no longer equal to Π^*. In fact, Roberts and Tweedie [166] showed that if the step size η is not chosen carefully, then the Markov chain defined by equation (2.10) can become transient and have no stationary distribution. However, in a series of recent works [43, 55, 38], it has been established that with a careful choice of step-size η and iteration count K, running the chain (2.10) for exactly K steps yields an iterate x_K whose distribution is close to Π^*. This more recent body of work provides non-asymptotic bounds that explicitly quantify the rate of convergence for this chain. Note that the algorithm (2.10) does not belong to the class of Metropolis-Hastings algorithm, since it does not involve an accept-reject step and does not have the target distribution Π^* as its stationary distribution. For these reasons, in the literature, this algorithm is referred to as the *unadjusted Langevin Algorithm*, or ULA for short.

Metropolis adjusted Langevin algorithm An alternative approach to handling the discretization error is to adopt $\mathcal{N}(x_k - \eta \nabla f(x_k), 2\eta \mathbb{I}_d)$ as the proposal distribution, and perform the Metropolis-Hastings accept-reject step. Doing so leads to the *Metropolis-adjusted Langevin Algorithm*, or MALA for short. We describe the different steps of MALA in Algorithm 2. As mentioned earlier, the Metropolis-Hastings correction ensures that the distribution of the MALA iterates $\{x_k\}$ converges to the correct distribution Π^* as $k \to \infty$. Indeed, since at each step the chain can reach any state $x \in \mathbb{R}^d$, it is strongly Π^*-irreducible and thereby ergodic [135, 51].

Both MALA and ULA are instances of first order sampling methods since they make use of both the function and the gradient values of f at different points. A natural question is if employing the accept-reject step for the discretization (2.10) provides any gain in the convergence rate. Our analysis to follow answers this question in the affirmative.

2.3 Main convergence results

We now state our main results for mixing time bounds for MALA and MRW. The overview of our results is as follows: First, we discuss the case of strongly log-concave

Algorithm 2: Metropolis adjusted Langevin algorithm (MALA)

Input: Step size η and a sample x_0 from a starting distribution μ_0

Output: Sequence $x_1, x_2, \ldots$

for $i = 0, 1, \ldots$ **do**

  Proposal step: *Draw* $z_{i+1} \sim \mathcal{N}(x_i - \eta \nabla f(x_i), 2\eta \mathbb{I}_d)$

  Accept-reject step:

    compute

$$\alpha_{i+1} \leftarrow \min\left\{1, \frac{\exp\left(-f(z_{i+1}) - \|x_i - z_{i+1} + \eta \nabla f(z_{i+1})\|_2^2 / 4\eta\right)}{\exp\left(-f(x_i) - \|z_{i+1} - x_i + \eta \nabla f(x_i)\|_2^2 / 4\eta\right)}\right\}$$

    With probability α_{i+1} *accept* the proposal: $x_{i+1} \leftarrow z_{i+1}$

    With probability $1 - \alpha_{i+1}$ *reject* the proposal: $x_{i+1} \leftarrow x_i$

end

densities and state the results for the two random walks from a warm start in Section 2.3.2 and from certain feasible starting distributions in Section 2.3.3, and then we consider the case of weakly log-concave densities.

2.3.1 Regularity conditions

We focus on the case when the negative log density $f(x) := -\log \Pi^*(x)$ is smooth and strongly convex. Recall from Section 1.5 that a function f is said to be L-smooth if

$$f(y) - f(x) - \nabla f(x)^\top (y - x) \leq \frac{L}{2} \|x - y\|_2^2 \quad \text{for all } x, y \in \mathbb{R}^d. \tag{2.11a}$$

In the other direction, a convex function f is said to be m-strongly convex if

$$f(y) - f(x) - \nabla f(x)^\top (y - x) \geq \frac{m}{2} \|x - y\|_2^2 \qquad \text{for all } x, y \in \mathbb{R}^d. \tag{2.11b}$$

The rates derived in this chapter apply to log-concave distributions given by equation (2.3) such that f is continuously differentiable on $\mathbb{R}^d$, and is both L-smooth and m-strongly convex. For such a function f, its condition number κ is defined as $\kappa := L/m$. We also refer to κ as the condition number of the distribution Π^*.

We summarize the mixing time bounds of several sampling algorithms in Tables 2.1 and 2.2, as a function of the dimension d, the error-tolerance ϵ, and the condition number κ. In Table 2.1, we state the results when the chain has a warm-start defined below (refer to Definition 1). Table 2.2 summarizes mixing time bounds from a particular distribution $\mu_\dagger$. Furthermore, in Section 2.3.4 we discuss the case when the f is smooth but not strongly convex and show that a suitable adaptation of MALA has a faster mixing rate compared to ULA for this case.

Random walk	Strongly log-concave	Weakly log-concave
ULA [38]	$\mathcal{O}\left(\frac{d\kappa^2 \log((\log \varpi)/\epsilon)}{\epsilon^2}\right)$	$\tilde{\mathcal{O}}\left(\frac{dL^2}{\epsilon^6}\right)$
ULA [43]	$\mathcal{O}\left(\frac{d\kappa^2 \log^2(\varpi/\epsilon)}{\epsilon^2}\right)$	$\tilde{\mathcal{O}}\left(\frac{d^3L^2}{\epsilon^4}\right)$
MRW	$\mathcal{O}\left(d\kappa^2 \log\left(\frac{\varpi}{\epsilon}\right)\right)$	$\tilde{\mathcal{O}}\left(\frac{d^3\,L^2}{\epsilon^2}\right)$
MALA	$\mathcal{O}\left(\max\left\{d\kappa, d^{0.5}\kappa^{1.5}\right\} \log\left(\frac{\varpi}{\epsilon}\right)\right)$	$\tilde{\mathcal{O}}\left(\frac{d^2\,L^{1.5}}{\epsilon^{1.5}}\right)$

Table 2.1. Scalings of upper bounds on ϵ-mixing time for different random walks in $\mathbb{R}^d$ with target $\Pi^* \propto e^{-f}$. In the second column, we consider smooth and strongly log-concave densities, and report the bounds from a ϖ-warm start for densities such that $m\mathbb{I}_d \preceq \nabla^2 f(x) \preceq L\mathbb{I}_d$ for any $x \in \mathbb{R}^d$ and use $\kappa := L/m$ to denote the condition number of the density. The big-O notation hides universal constants. We remark that the presented bounds for ULA in this column are not stated in the corresponding papers, and are derived by us, using their framework. In the last column, we summarize the scaling for weakly log-concave smooth densities: $0 \preceq \nabla^2 f(x) \preceq L\mathbb{I}_d$ for all $x \in \mathbb{R}^d$. For this case, the $\tilde{O}$ notation is used to track scaling only with respect to d, ϵ and L and ignore dependence on the starting distribution and a few other parameters.

2.3.2 Mixing time bounds for warm start

In the analysis of Markov chains, it is convenient to have a rough measure of the distance between the initial distribution μ_0 and the stationary distribution. As in past work on the problem, we adopt the following notion of *warmness*:

Definition 1 (Warm start)**.** *For a finite scalar $\varpi > 0$, the initial distribution μ_0 is said to be ϖ-warm with respect to the stationary distribution Π^* if*

$$\sup_A \left(\frac{\mu_0(A)}{\Pi^*(A)}\right) \leq \varpi, \tag{2.12}$$

where the supremum is taken over all measurable sets A.

In parts of our work, we provide bounds on the quantity

$$\tau_1(\epsilon; \varpi) = \sup_{\mu_0 \in \mathcal{P}_\varpi(\Pi^*)} \tau_1(\epsilon; \mu_0)$$

where $\mathcal{P}_\varpi(\Pi^*)$ denotes the set of all distributions that are ϖ-warm with respect to Π^*. Naturally, as the value of ϖ decreases, the task of generating samples from the target distribution becomes easier.[2] However, access to a good "warm" distribution (small

[2]For instance, $\varpi = 1$ implies that the chain starts at the stationary distribution and has already mixed.

Random walk	$\mu_\dagger$	$\tau_1(\epsilon;\mu_0)$
ULA [38]	$\mathcal{N}(x^\star, m^{-1}\mathbb{I}_d)$	$\frac{d\kappa^2\log(d\kappa/\epsilon)}{\epsilon^2}$
ULA [43]	$\mathcal{N}(x^\star, L^{-1}\mathbb{I}_d)$	$\frac{(d^3+d\log^2(1/\epsilon))\kappa^2}{\epsilon^2}$
MRW	$\mathcal{N}(x^\star, L^{-1}\mathbb{I}_d)$	$d^2\kappa^2\log^{1.5}\left(\frac{\kappa}{\epsilon}\right)$
MALA	$\mathcal{N}(x^\star, L^{-1}\mathbb{I}_d)$	$d^2\kappa\log\left(\frac{\kappa}{\epsilon}\right)$

Table 2.2. Scalings of upper bounds on ϵ-mixing time, from the starting distribution $\mu_\dagger$ given in column two, for different random walks in $\mathbb{R}^d$ with target $\Pi^* \propto e^{-f}$ such that $m\mathbb{I}_d \preceq \nabla^2 f(x) \preceq L\mathbb{I}_d$ for any $x \in \mathbb{R}^d$ and $\kappa := L/m$. Here $x^\star$ denotes the unique mode of the target density Π^*.

ϖ) may not be feasible for many applications, and thus deriving bounds on mixing time of the Markov chain from non-warm starts is also desirable. Consequently, in the sequel, we also provide practical initialization methods and polynomial-time mixing time guarantees from such starts.

Our mixing time bounds involve the functions r and w given by

$$r(s) = 2 + 2\cdot\max\left\{\frac{1}{d^{0.25}}\log^{0.25}\left(\frac{1}{s}\right), \frac{1}{d^{0.5}}\log^{0.5}\left(\frac{1}{s}\right)\right\}, \quad \text{and} \tag{2.13a}$$

$$w(s) = \min\left\{\frac{\sqrt{m}}{r(s)\cdot L\sqrt{dL}}, \frac{1}{Ld}\right\} \qquad \text{for } s \in \left(0, \tfrac{1}{2}\right). \tag{2.13b}$$

We use $\mathcal{T}_{\mathrm{MALA}(\eta)}$ to denote the transition operator on probability distributions induced by one step of MALA. We have the following mixing time bound for the MALA algorithm for a strongly-log concave measure from a warm start.

Theorem 1. *For any ϖ-warm initial distribution μ_0 and any error tolerance $\epsilon \in (0,1]$, the Metropolis adjusted Langevin algorithm with step size $\eta = c\,w(\epsilon/(2\varpi))$ satisfies $d_{TV}\left(\mathcal{T}^k_{MALA(\eta)}(\mu_0), \Pi^*\right) \le \epsilon$ for all*

$$k \ge c'\log\left(\frac{2\varpi}{\epsilon}\right)\max\left\{d\kappa,\ d^{0.5}\kappa^{1.5}r\left(\frac{\epsilon}{2\varpi}\right)\right\}, \tag{2.14}$$

where c, c' denote universal constants.

See Section 2.5.2 for the proof.

Note that $r(s) \leq 4$ for $s \geq e^{-d}$ and thus we can treat $r(\epsilon/2\varpi)$ as small constant for most interesting values of ϵ if the warmness parameter ϖ is not too large. Consequently, we can run MALA with a fixed step size η for a large range of error-tolerance ϵ. Treating $r(\cdot)$ as a constant, we obtain that if $\kappa = o(d)$, the mixing time of MALA scales as $O\left(d\kappa \log(1/\epsilon)\right)$ which is exponentially better in the tolerance-ϵ compared to $O\left(d\kappa^2 \log^2(1/\epsilon)/\epsilon^2\right)$ mixing time of ULA, and has better dependence on κ while still maintaining linear dependence on d. In fact, for any setting of κ, d and ϵ, MALA always has a better mixing time bound compared to ULA. A limitation of our analysis is that the constant c' is not small. However we demonstrate in Section 2.4 that in practice small constants provide performance that match the scalings suggested by our theoretical bounds.

Let $\mathcal{T}_{\mathrm{MRW}(\eta)}$ denote the transition operator on the space of probability distributions induced by one step of MRW. We now state the convergence rate for Metropolized random walk for strongly-log concave density.

Theorem 2. *For any ϖ-warm initial distribution μ_0 and any $\epsilon \in (0, 1]$, the Metropolized random walk with step size $\eta = \frac{cm}{dL^2 r(2\varpi)}$ satisfies*

$$d_{TV}\left(\mathcal{T}^k_{MRW(\eta)}(\mu_0), \Pi^*\right) \leq \epsilon \quad \textit{for all} \quad k \geq c'\, d\kappa^2 r\left(\frac{\epsilon}{2\varpi}\right) \log\left(\frac{2\varpi}{\epsilon}\right), \tag{2.15}$$

where c, c' denote universal constants.

See Section 2.5.6 for the proof.

Again treating $r(\epsilon/2\varpi)$ as a small constant, we find that the mixing time of MRW scales as $O\left(d\kappa^2 \log(1/\epsilon)\right)$ which has an exponential factor in ϵ better than ULA. Compared to the mixing time bound for MALA, the bound in Theorem 2 has an extra factor of $O(\kappa)$. While such a factor is conceivable given that MALA's proposal distribution uses first order information about the target distribution, and MRW uses only the function values, it would be interesting to determine if this gap can be improved in a future work.

2.3.3 Mixing time bounds for a feasible start

In many cases, a good warm start may not be available. Consequently, mixing time bounds from a feasible starting distribution can be useful in practice. Letting $x^\star$ denote the unique mode of the distribution Π^*, we claim that the distribution $\mu_\dagger = \mathcal{N}(x^\star, L^{-1}\mathbb{I}_d)$ is one such choice. Recalling the notation $\kappa = L/m$, we claim that the warmness parameter for $\mu_\dagger$ can be bounded as follows:

$$\sup_A \frac{\mu_\dagger(A)}{\Pi^*(A)} \leq \kappa^{d/2} = \varpi_\star, \tag{2.16}$$

where the supremum is taken over all measurable sets A. When the gradient ∇f is available, finding $x^\star$ comes at nominal additional cost: in particular, standard optimization algorithms such as gradient descent be used to compute a δ-approximation of $x^\star$ in $O\kappa \log(1/\delta)$ steps (e.g., see the monograph [24]). Also refer to Section 2.3.3 for more details when we have inexact parameters.

Assuming claim (2.16) for the moment, we now provide mixing time bounds for MALA and MRW with $\mu_\dagger$ as the starting distribution. For any threshold $\epsilon \in (0, 1]$, we define the step sizes $\eta_1 = c' w(\epsilon/2\varpi_\star)$ and $\eta_2 = \frac{c'm}{dL^2 \cdot r(\epsilon/2\varpi_\star)}$, where the function w was previously defined in equation (2.13b).

Corollary 1. *With $\mu_\dagger$ as the starting distribution, we have*

$$d_{TV}\left(\mathcal{T}^k_{MRW(\eta_2)}(\mu_\dagger), \Pi^*\right) \leq \epsilon \quad \text{for all } k \geq c\, d^2\kappa^2 \log^{1.5}\left(\frac{\kappa}{\epsilon^{1/d}}\right), \quad \text{and} \tag{2.17a}$$

$$d_{TV}\left(\mathcal{T}^k_{MALA(\eta_1)}(\mu_\dagger), \Pi^*\right) \leq \epsilon \quad \text{for all } k \geq c\, d^2\kappa \log\left(\frac{\kappa}{\epsilon^{1/d}}\right) \max\left\{1, \sqrt{\frac{\kappa}{d}\log\left(\frac{\kappa}{\epsilon^{1/d}}\right)}\right\}. \tag{2.17b}$$

The proof follows by plugging the bound (2.16) in Theorem 1 and 2 and is thereby omitted.

We now prove the claim (2.16). Without loss of generality, we can assume that $f(x^\star) = 0$. Such an assumption is possible because substituting $f(\cdot)$ by $f(\cdot) + \alpha$ for any scalar α leaves the distribution Π^* unchanged. Since f is m-strongly convex and L-smooth, we obtain that

$$\frac{L}{2}\|x - x^\star\|_2^2 \geq f(x) \geq \frac{m}{2}\|x - x^\star\|_2^2.$$

Consequently, we find that $\int_{\mathbb{R}^d} e^{-f(x)}dx \leq (2\pi/m)^{d/2}$. Making note of the lower bound

$$\Pi^*(x) \geq \frac{e^{-\frac{L}{2}\|x-x^\star\|_2^2}}{(2\pi m^{-1})^{d/2}}, \tag{2.18}$$

and plugging in the expression for the density of $\mu_\dagger$ yields the claim (2.16).

We now derive results for the case when we do not have access to exact parameters, e.g., if the mode $x^\star$ is known approximately, and/or we only have an upper bound for the smoothness parameter L—a situation quite prevalent in practice.

Starting distribution with inexact parameters

Note that $x^\star$ is also the unique global minima of the negative log-density f. For the strongly convex function f, using a first-order method, like gradient descent, we can obtain an ε-approximate mode $\tilde{x}$ using $\kappa \log(1/\varepsilon)$ evaluations of the gradient ∇f. Suppose we have access to a point $\tilde{x}$ such that $\|\tilde{x} - x^\star\|_2 \leq \varepsilon$ and have an upper bound estimate $\tilde{L} \geq L$ for the smoothness.

We now consider the case of starting distribution $\tilde{\mu} = \mathcal{N}(\tilde{x}, (2\tilde{L})^{-1}\mathbb{I}_d)$, as a proxy for the feasible start $\mu_\dagger = \mathcal{N}(x^\star, L^{-1}\mathbb{I}_d)$ discussed above. Note the difference in mean and the covariance between the distributions $\tilde{\mu}$ and $\mu_\dagger$. Given the handy result in Theorem 1, it suffices to bound the warmness parameter for the distribution $\tilde{\mu}$. Applying triangle inequality, we obtain that

$$\|x - \tilde{x}\|_2^2 \geq \frac{1}{2}\|x - x^\star\|_2^2 - \|x^\star - \tilde{x}\|_2^2 \tag{2.19}$$

and consequently that

$$\begin{aligned}\tilde{\mu}(x) &= (\pi\tilde{L}^{-1})^{-d/2} \exp\left(-\tilde{L}\|x - \tilde{x}\|_2^2\right) \\ &\leq (\pi\tilde{L}^{-1})^{-d/2} \exp\left(-\frac{\tilde{L}\|x - x^\star\|_2^2}{2} + \tilde{L}\|\tilde{x} - x^\star\|_2^2\right)\end{aligned}$$

Using the lower bound (2.18) on the target density, we find that

$$\begin{aligned}\frac{\tilde{\mu}(x)}{\Pi^*(x)} &\leq \left(\frac{\tilde{L}}{L} \cdot 2\kappa\right)^{d/2} \exp\left(\tilde{L}\|\tilde{x} - x^\star\|_2^2 - \frac{(\tilde{L} - L)\|x - x^\star\|_2^2}{2}\right) \\ &\leq \exp\left(\frac{d}{2}\log(2\kappa\tilde{L}/L) + \tilde{L}\varepsilon^2\right),\end{aligned}$$

where the last inequality follows from the fact that $\tilde{L} \geq L$. In other words, the distribution $\tilde{\mu}$ is $\tilde{\varpi}$-warm with respect to the target distribution Π^*, where we define $\tilde{\varpi} = \exp\left(\frac{d}{2}\log(2\kappa\tilde{L}/L) + \tilde{L}\varepsilon^2\right)$.

Using Theorem 1, we now derive a mixing time bound for MALA with starting distribution $\tilde{\mu}$. For any threshold $\epsilon \in (0, 1]$, we use the step size $\eta_3 = c'w(\epsilon/(2\tilde{\varpi}))$. Invoking Theorem 1 and plugging in the definition (2.13a) of w, we find that the total variation distance satisfies $d_{\mathrm{TV}}\left(\mathcal{T}^k_{\mathrm{MALA}(\eta_3)}(\tilde{\mu}), \Pi^*\right) \leq \epsilon$, for all

$$k \geq cd^2\kappa\left(\log\frac{2\kappa\tilde{L}/L}{\epsilon^{1/d}} + \frac{\tilde{L}\varepsilon^2}{d}\right)\max\left\{1, \sqrt{\frac{\kappa}{d}}\left(\sqrt{\log\frac{2\kappa\tilde{L}/L}{\epsilon^{1/d}}} + \frac{\sqrt{\tilde{L}}\varepsilon}{\sqrt{d}}\right)\right\}, \tag{2.20}$$

which also recovers the bound from corollary 1 for MALA as $\varepsilon \to 0$ and $\tilde{L} \to L$. Note that the mixing time increases (additively) by $O\left(\kappa d\varepsilon^2\tilde{L}/L\right)$ when we only have an ε-approximate mode, which is an $(\tilde{L}/L \cdot \varepsilon/d)$-fraction increase in the mixing time bound with starting distribution $\mu_\dagger$. A mixing time bound for MRW with starting distribution $\tilde{\mu}$ can be obtained in a similar fashion and is thereby omitted.

2.3.4 Weakly log-concave densities

In this section, we show that MALA can also be used for approximate sampling from a density that is L-smooth and (weakly) log-concave, but not necessary strongly log-concave. The key idea is to approximate the given log-concave density Π^* with a strongly log-concave density $\widetilde{\pi}^*$ such that the total variation distance $d_{\mathrm{TV}}\left(\widetilde{\pi}^*, \Pi^*\right)$ is small. Next, we use MALA to sample from $\widetilde{\pi}^*$ and consequently obtain an approximate sample from Π^*. In order to construct $\widetilde{\pi}^*$, we use a scheme previously suggested by Dalalyan [43]. With λ as a tuning parameter, consider the distribution $\widetilde{\pi}^*$ given by the density

$$\widetilde{\pi}^*(x) = \frac{1}{\displaystyle\int_{\mathbb{R}^d} e^{-\widetilde{f}(y)} dy} e^{-\widetilde{f}(x)} \quad \text{where} \quad \widetilde{f}(x) = f(x) + \frac{\lambda}{2}\left\|x - x^\star\right\|_2^2. \tag{2.21}$$

Dalalyan (Lemma 3 in the paper [43]) showed that that the total variation distance between Π^* and $\widetilde{\pi}^*$ is bounded as follows:

$$d_{\mathrm{TV}}\left(\widetilde{\pi}^*, \Pi^*\right) \leq \frac{1}{2}\left\|\widetilde{f} - f\right\|_{L^2(\Pi^*)} \leq \frac{\lambda}{4}\left(\int_{\mathbb{R}^d} \left\|x - x^\star\right\|_2^4 \Pi^*(x) dx\right)^{1/2}.$$

Suppose that the original distribution Π^* has its fourth moment bounded as

$$\int_{\mathbb{R}^d} \left\|x - x^\star\right\|_2^4 \Pi^*(x) dx \leq d^2\nu^2. \tag{2.22}$$

We now set $\lambda := 2\epsilon/(d\nu)$ to obtain $d_{\mathrm{TV}}\left(\widetilde{\pi}^*, \Pi^*\right) \leq \epsilon/2$. Since $\widetilde{f}$ is $\lambda/2$-strongly convex and $L + \lambda/2$-smooth, the condition number of $\widetilde{\pi}^*$ is given by $\tilde{\kappa} = 1 + Ld\nu/\epsilon$. We substitute $\tilde{\kappa} = Ld\nu/\epsilon$ to obtain simplified expressions for mixing time bounds in the results that follow. Since now the target distribution is $\widetilde{\pi}^*$, we suitably modify the step size for MALA as follows:

$$w_{\mathrm{lc}}(s) = \frac{1}{Ld} \min\left\{\frac{\sqrt{s}}{r(s)\sqrt{\nu L}},\ 1\right\}$$

where the function r was previously defined in equation (2.13a). We refer to this new set-up with a modified target distribution $\widetilde{\pi}^*$ as the *modified MALA method*. To keep our results simple to state, we assume that we have a warm start with respect to $\widetilde{\pi}^*$.

Corollary 2. *Assume that Π^* satisfies (2.22). Then for any given error-tolerance $\epsilon \in (0,1)$, and, any ϖ-warm start μ_0, the modified MALA method with step size $\eta = c w_{lc}(\epsilon/(2\varpi))$ satisfies $d_{TV}\left(\mathcal{T}^k_{MALA(\eta)}(\mu_0), \Pi^*\right) \leq \epsilon$ for all*

$$k \geq c' \log\left(\frac{4\varpi}{\epsilon}\right) \max\left\{\frac{d^2 L\nu}{\epsilon},\ d^2\left(\frac{L\nu}{\epsilon}\right)^{1.5} r\left(\frac{\epsilon}{4\varpi}\right)\right\},$$

where c, c' denote universal positive constants.

The proof follows by combining the triangle inequality, as applied to the TV norm, along with the bound from Theorem 1. Thus, for weakly log-concave densities, modified MALA mixes in $O\left(d^2/\epsilon^{1.5}\right)$, which improves upon the $O\left(d^3/\epsilon^4\right)$ mixing time bound for a ULA scheme on $\widetilde{\pi}^*$, as established by Dalalyan [43]. A mixing time bound of $O\left(d^3/\epsilon^2\right)$ for MRW can be derived similarly for this case, simply by noting that the new condition number $\tilde{\kappa} = Ld\nu/\epsilon$ for the modified density and the fact that the mixing time of MRW is $O\left(d\tilde{\kappa}^2\right)$ in the strongly log-concave setting.

2.4 Numerical experiments

In this section, we compare MALA with ULA and MRW in various simulation settings. The step-size choice of ULA follows from [43] in the case of warm start. The step-size choice of MALA and MRW used in our experiments in our results are summarized in Table 2.3. We consider four different experiments:

(i) sampling multivariate Gaussian

(ii) sampling from a mixture of two Gaussian distributions

(iii) estimating the MAP with credible intervals in a Bayesian logistic regression set-up

(iv) simulating accept-reject rate based on step-size choices.

Since TV distance for continuous measures is hard to estimate, we use several proxy measures for convergence diagnostics:

(a) errors in quantiles

(b) ℓ_1-distance in histograms (discrete tv-error)

(c) error in sample MAP estimate

(d) trace-plot along different coordinates

(e) autocorrelation plot.

While the first three measures are useful for diagnosing the convergence of random walks over several independent runs, the last two measures are useful for diagnosing the rate of convergence of the Markov chain in a single long run.

2.4.1 Dimension dependence for multivariate Gaussian

The goal of this simulation is to demonstrate the dimension dependence in experiments, for mixing time of ULA, MALA and MRW when the target is non-isotropic multivariate Gaussian. Note that Theorem 1 and 2 imply that the dimension dependency for

both MALA and MRW is d. We consider sampling from multivariate Gaussian with density Π^* defined by

$$x \mapsto \Pi^*(x) \propto e^{-\frac{1}{2}x^\top \Sigma^{-1} x}, \tag{2.23}$$

where $\Sigma \in \mathbb{R}^{d\times d}$ the covariance matrix to be specified. For this target distribution, the function f, its derivatives are given by

$$f(x) = \frac{1}{2}x^\top \Sigma^{-1} x, \quad \nabla f(x) = \Sigma^{-1} x, \quad \text{and} \quad \nabla^2 f(x) = \Sigma^{-1}.$$

Consequently, the function f is strongly convex with parameter $m = 1/\lambda_{\max}(\Sigma)$ and smooth with parameter $L = 1/\lambda_{\min}(\Sigma)$. For convergence diagnostics, we use the error in quantiles along different directions. Using the exact quantile information for each direction for Gaussians, we measure the error in the 75% quantile of the sample distribution and the true distribution in the *least favorable direction*, i.e., along the eigenvector of Σ corresponding to the eigenvalue $\lambda_{\max}(\Sigma)$. The approximate mixing time is defined as the smallest iteration when this error falls below δ. We use $\mu_\dagger$ as the initial distribution where $\mu_\dagger = \mathcal{N}(0, L^{-1}\mathbb{I}_d)$.

Strongly log-concave density

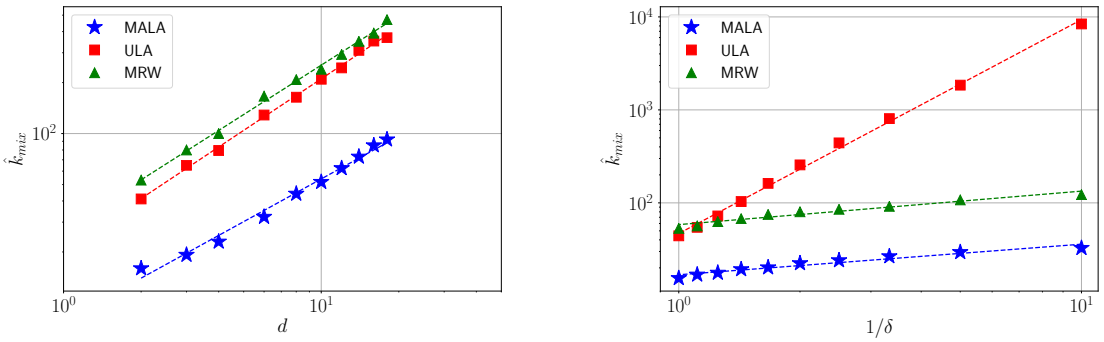
The step-sizes are chosen according to Table 2.3. For ULA, the error-tolerance ϵ is chosen to be 0.2. We set Σ as a diagonal matrix with the largest eigenvalue 4.0 and the smallest eigenvalue 1.0 so that the $\kappa = 4$ is fixed across different settings. For a fixed dimension d, we simulate 10 independent runs of the three chains each with $N = 10,000$ samples to determine the approximate mixing time. The final approximate mixing time for each walk is the average of that over these 10 independent runs. Figure 2.1(a) shows the dependency of the approximate mixing time as a function of dimension d for the three random walks in log-log scale. To examine the dimension dependency, we perform linear regression for approximate mixing time with respect to dimensions in the log-log scale. The computations reveal that the dimension dependency of MALA, ULA and MRW are all close to order d (slope 0.84, 1.01 and 0.97). Figure 2.1(b) shows the dependency of the approximate mixing time on the inverse error $1/\epsilon$ for the three random walks in log-log scale. For ULA, the step-size is error-dependent, precisely chosen to be 10 times of ϵ. A linear regression of the approximate mixing time on the inverse error $1/\epsilon$ yields a slope of 2.23 suggesting the error dependency of order $1/\epsilon^2$ for ULA. A similar computation for MALA and MRW yields a slope of 0.33 for both the cases which not only suggests a significantly better error dependency for these two chains but also partly verifies their theoretical mixing time bounds of order $\log(1/\epsilon)$.

Weakly log-concave density

We now discuss the convergence of the random walks when the Gaussian is flat along a direction. In particular, we consider the Gaussian distribution such that $\lambda_{\max}(\Sigma) =$

Random walk	ULA	MALA	MRW
Step size	$\frac{\epsilon^2}{d\kappa L}$	$\frac{1}{L}\min\left\{\frac{1}{\sqrt{d\kappa}}, \frac{1}{d}\right\}$	$\frac{1}{d\kappa L}$

Table 2.3. Step-size used in simulations to obtain ϵ-accuracy for different random walks in $\mathbb{R}^d$ with target $\Pi^* \propto e^{-f}$ such that $m\mathbb{I}_d \preceq \nabla^2 f(x) \preceq L\mathbb{I}_d$ for any $x \in \mathbb{R}^d$ and $\kappa := L/m$.

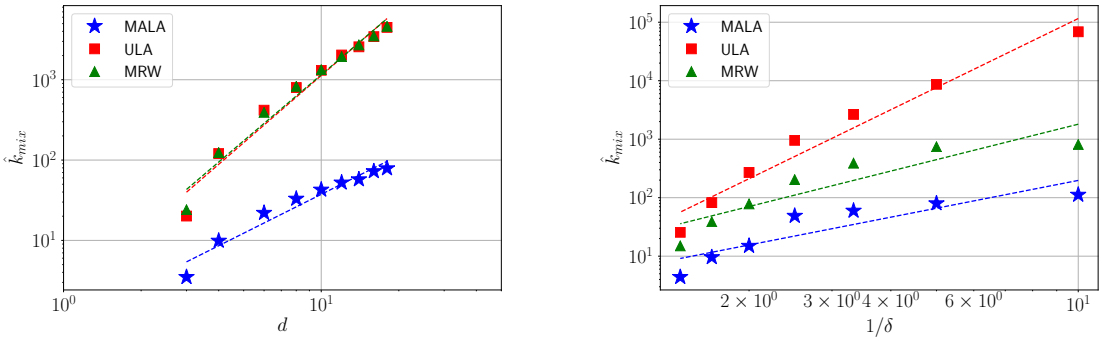


(a) (b)

Figure 2.1. Discrete TV error on Gaussian density (2.23) where the covariance has condition number $\kappa = 4$. (a) Dimension dependency. (b) Error-tolerance dependency.

1000 and $\lambda_{\min}(\Sigma) = 1$. Such a setting implies that the strong convexity parameter $m = 0.001$ and our target density mimics a weakly log-concave density. For convergence diagnostics, we use the error in quantiles along one direction other than the ones which correspond to $\lambda_{\max}(\Sigma)$ and $\lambda_{\min}(\Sigma)$. Using the exact quantile information for each direction for Gaussians, we measure the error between the 75% quantile of the sample distribution and the true distribution in that direction. The approximate mixing time is defined as the smallest iteration when this error falls below δ. We use $\mu_\dagger$ as the initial distribution where $\mu_\dagger = \mathcal{N}(0, L^{-1}\mathbb{I}_d)$. The step-sizes are chosen according to Table 2.3 where m is chosen to be $\epsilon/(dL)$. For dimension dependence experiments, we fix the error-tolerance ϵ as 0.2. For a fixed dimension d, we simulate 10 independent runs of the three chains each with $N = 10,000$ samples to determine the approximate mixing time. The final approximate mixing time for each walk is the average of that over these 10 independent runs. Figure 2.2(a) and 2.2(b) show the dependency of the approximate mixing time as a function of dimension d and the inverse error $1/\epsilon$ respectively, for the three random walks on this weakly log-concave density (log-log scale). Linear fits on the log-log scale reveal that the dimension dependence of mixing time for MALA is close to d^2 (slope 1.61), and that for ULA is close to d^3 (slope 2.78) and for MRW it is approximately of order d^3 (slope 2.73). Linear fits of the approximate mixing time on the inverse error $1/\epsilon$ yield a slope of 3.92 for ULA thereby suggesting an error

dependence of order $1/\epsilon^4$, while for MALA and MRW this dependence is of order $1/\epsilon^{1.5}$ (slope 1.56) and of order $1/\epsilon^2$ (slope 2.01), respectively. These scalings partly verify the rates derived in Corollary 2 and demonstrate the gains of MALA over ULA for the weakly log-concave densities.



(a) (b)

Figure 2.2. Scaling of mixing times from weakly log-concave Gaussian density. (a) Dimension dependency. (b) Error-tolerance dependency.

Warmness in simulations

Strictly speaking, for both the cases considered above, the starting distribution was not warm, since we used $\mu_\dagger$ as the starting distribution and the corresponding warmness $\varpi = O(e^d)$ scales exponentially with dimension d. However, the mixing time observed in the simulations, albeit with a heuristic measure, are d times faster than those stated with $\mu_\dagger$ as the starting distribution in Corollary 1, and are in fact consistent with the results for the warm-start which are stated in Theorems 1 and 2. We believe that the results stated in Corollary 1, with $\mu_\dagger$ as the starting distribution, can be improved by a factor of d. Even though the proof technique in this chapter does not close this gap, the techniques in Chapter 3 show that the dependency in warmness of the starting distribution is negligeable.

2.4.2 Behavior for Gaussian mixture distribution

We now consider the task of sampling from a two component Gaussian mixture distribution, as previously considered by Dalalyan [43] for illustrating the behavior of ULA. Here compare the behavior of MALA to ULA for this case. The target density is given by

$$x \mapsto \Pi^*(x) = \frac{1}{2\,(2\pi)^{d/2}} \left(e^{-\|x-a\|_2^2/2} + e^{-\|x+a\|_2^2/2} \right),$$

where $a \in \mathbb{R}^d$ is a fixed vector. This density corresponds to the two-mixture of equal weighted Gaussians $\mathcal{N}(a, \mathbb{I}_d)$ and $\mathcal{N}(-a, \mathbb{I}_d)$. In our notation, the function f and its derivatives are given by: $f(x) = \frac{1}{2}\|x-a\|_2^2 - \log(1+e^{-2x^\top a})$,

$$\nabla f(x) = x - a + 2a(1+e^{2x^\top a})^{-1}, \text{ and } , \nabla^2 f(x) = \mathbb{I}_d - 4aa^\top \frac{e^{2x^\top a}}{\left(1+e^{2x^\top a}\right)^2}.$$

From examination of the Hessian, we see that the function f is smooth with parameter $L = 1$, and whenever $\|a\|_2 < 1$, it is also strongly convex with parameter $m = 1 - \|a\|_2^2$.

For dimension $d = 2$, setting $a = \left(\frac{1}{2}, \frac{1}{2}\right)$ yields the parameters $m = \frac{1}{2}$ and $L = 1$. Figure 2.3 shows the level sets of the density of this 2D-Gaussian mixture. The initial distribution is chosen as $\mu_\dagger = \mathcal{N}\left(0, L^{-1}\mathbb{I}_d\right)$ and the step-sizes are chosen according to Table 2.1, where for ULA, we set three different choices of $\epsilon = 0.2$ (ULA), $\epsilon = 0.1$ (small-step ULA) and $\epsilon = 1.0$ (large-step ULA). Note that choosing a smaller threshold ϵ implies that the ULA has a smaller step size and consequently the chain takes larger to converge. However, the asymptotic TV error with respect to the target distribution Π^* for ULA also decreases with decrease in step size. These different choices of step sizes are made to demonstrate the fundamental trade-off between the rate of convergence and asymptotic error for ULA and its inability to mix faster than MALA for different settings.

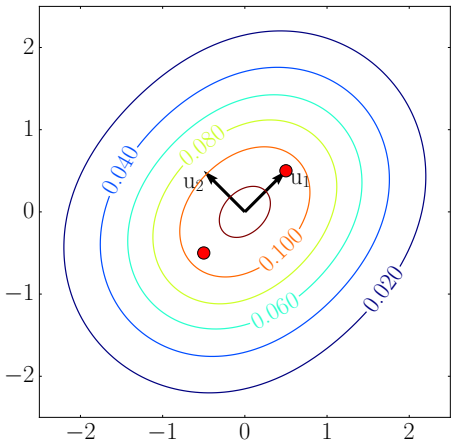


Figure 2.3. Level set of the density of the 2D Gaussian mixture. The red dots are the location of the means a and $-a$, where a is chosen such that $\|a\|_2^2 = \frac{1}{2}$. The arrows indicate the two principal directions u_1 and u_2 along which the TV error is measured.

Note that one can sample directly from the mixture of Gaussian in consideration by drawing independently a Bernoulli(1/2) random variable y and a standard normal variable $z \sim \mathcal{N}(0, \mathbb{I}_d)$, and by computing

$$x = y \cdot (z - a) + (1 - y) \cdot (z + a)$$

This observation makes it easy to diagnose the convergence of our Markov chains with target Π^*. In order to estimate the total variation distance, we discretize the distribution of $N = 250,000$ samples from Π^* over a set of bins, and consider the total variation of this discrete distribution from the empirical distribution of the Markov chain over these bins. We refer to this measure as the discretized TV error. We measure the sum of two discrete TV errors of $250,000$ samples from Π^* with the empirical distribution obtained by simulating the chains ULA, MALA or MRW, projected on two principal directions (u_1 and u_2), over a discrete grid of size $B = 100$. Figure 2.4 shows the sum of the discretized TV errors along u_1 and u_2, as a function of iterations. The true total variation distance between the distribution of the iterate and the target distribution is upper bounded by the sum of (A) the discretized TV error and (B) the error caused by discretization. To obtain an idea of how large is the error (B) due to discretization, we simulate 100 runs of the discrete TV error between two independent drawings from the true distribution Π^*. The two black lines in Figure 2.4 are the maximum and minimum of these 100 values. The sample distribution at convergence is expected to lie between the two black lines.

Figure 2.4(a) shows that ULA converges significantly slower than MALA to the right distribution. Figure 2.4(b) illustrates this point further and shows that when compared to the ULA, the small-step ULA ($\epsilon = 0.1$) converges at a much slower rate and large-step ULA ($\epsilon = 1.0$) has a larger approximation error (asymptotic bias).

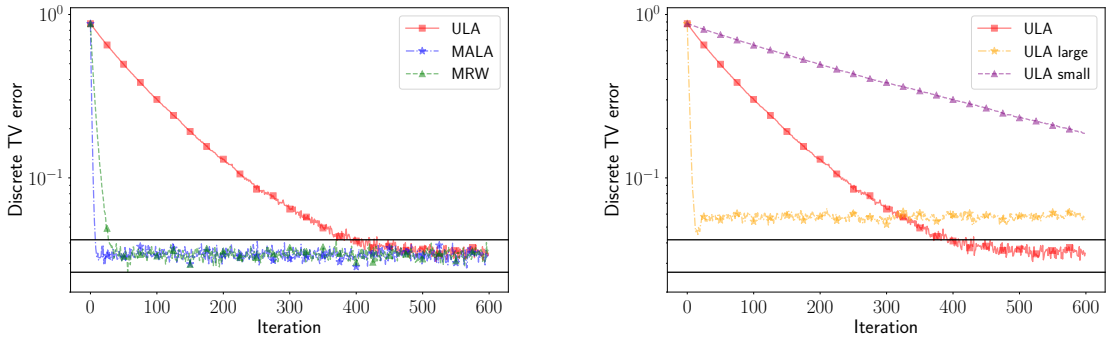
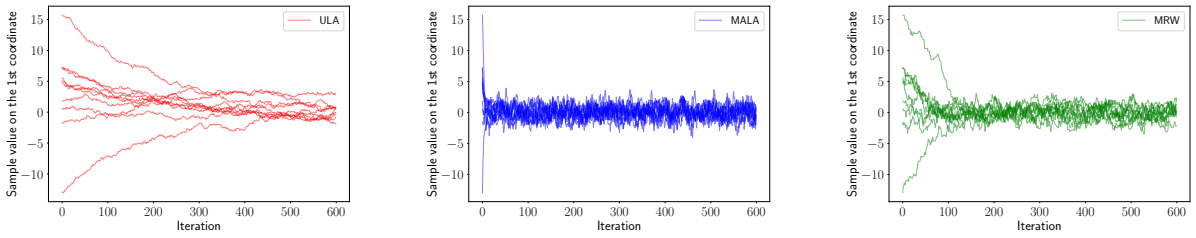


Figure 2.4. Discrete TV error on a two component Gaussian mixture. (a) Behavior of three different random walks. (b) Behavior of ULA with different choices of step sizes.

We accompany the study based on exact TV error computation with two classical convergence diagnostic plots for general MCMC algorithms. Figure 2.5 shows the traceplots of the three sampling algorithms in 10 runs. Comparing the three plots (Figure 2.5 (a), (b), (c)), we observe that the traceplot of MALA stabilizes much faster than that of ULA and MRW. Furthermore, to compare the efficiency of the chains in stationarity, Figure 2.6 shows the autocorrelation function of the three chains. To make sure that

the computation is done in stationarity, we set in practice the burn-in period to be 300 iterations. Again, we observe that MALA is clearly significantly more efficient than ULA and MRW.



(a) (b) (c)

Figure 2.5. Traceplot of the first coordinate on a two component Gaussian mixture. (a) Traceplot of ULA. (b) Traceplot of MALA. (c) Traceplot of MRW.

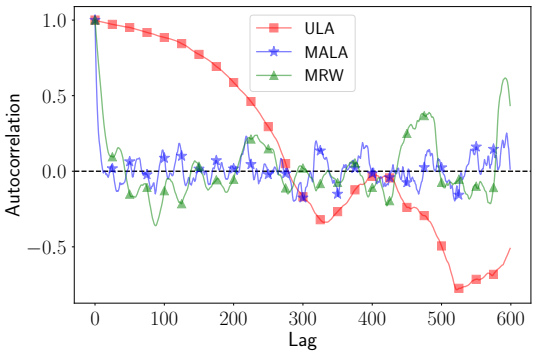


Figure 2.6. Markov chain autocorrelation function plot. The burn-in time for the plot is set to 300 iterations.

2.4.3 Bayesian Logistic Regression

We now consider the problem of logistic regression in a frequentist-Bayesian setting, similar to that considered by Dalalyan [43]. Once again, we establish that MALA has superior performance relative to ULA. Given a binary variable $y \in \{0, 1\}$ and a covariate $x \in \mathbb{R}^d$, the logistic model for the conditional distribution of y given x takes the form

$$\mathbb{P}(y = 1|x; \theta) = \frac{e^{\theta^\top x}}{1 + e^{\theta^\top x}}, \tag{2.24}$$

for some parameter $\theta \in \mathbb{R}^d$.

In a Bayesian framework, we model the parameter θ in the logistic equation as a random variable with a prior distribution Π_0^*. Suppose that we observe a set of independent samples $\{(x_i, y_i)\}_{i=1}^n$ with $(x_i, y_i) \in \mathbb{R}^d \times \{0, 1\}$, with each y_i conditioned on x_i drawn from a logistic distribution with some unknown parameter θ^*. Using Bayes' rule, we can then compute the posterior distribution of the parameter θ given the data. Drawing samples from this posterior distribution allows us to estimate and draw inferences about the unknown parameter. Under mild conditions, the Bernstein-von-Mises theorem guarantees that the posterior distribution will concentrate around the true parameter θ^*, in which case we expect that the credible intervals formed by sampling from the posterior should contain θ^* with high probability. This fact provides a lens for us to assess the accuracy of our sampling procedure.

Define the vector $Y = (y_1, \ldots, y_n)^\top \in \{0, 1\}^n$ and let X be the $n \times d$ matrix with x_i as i^{th}-row. We choose the prior π_0 to be a Gaussian distribution with zero mean and covariance matrix proportional to the inverse of the sample covariance matrix $\Sigma_X = \frac{1}{n} X^\top X$. Plugging in the formulas for the prior and likelihood, we find that the the posterior density is given by

$$\Pi^*(\theta) = \Pi^*(\theta | X, Y) \propto \exp\left\{ Y^\top X\theta - \sum_{i=1}^n \log\left(1 + e^{\theta^\top x_i}\right) - \alpha \left\| \Sigma_X^{1/2} \theta \right\|_2^2 \right\},$$

where $\alpha > 0$ is a user-specified parameter. Writing $\Pi^* \propto e^{-f}$, we observe that the function f and its derivatives are given by

$$\begin{aligned}
f(\theta) &= -Y^\top X\theta + \sum_{i=1}^n \log\left(1 + e^{\theta^\top x_i}\right) + \alpha \left\| \Sigma_X^{1/2} \theta \right\|_2^2, \\
\nabla f(\theta) &= -X^\top Y + \sum_{i=1}^n \frac{x_i}{1 + e^{-\theta^\top x_i}} + \alpha \Sigma_X \theta, \quad \text{and,} \\
\nabla^2 f(\theta) &= \sum_{i=1}^n \frac{e^{-\theta^\top x_i}}{\left(1 + e^{-\theta^\top x_i}\right)^2} x_i x_i^\top + \alpha \Sigma_X.
\end{aligned}$$

With some algebra, we can deduce that the eigenvalues of the Hessian $\nabla^2 f$ are bounded between $L := (0.25n + \alpha)\ \lambda_{\max}(\Sigma_X)$ and $m := \alpha\, \lambda_{\min}(\Sigma_X)$ where $\lambda_{\max}(\Sigma_X)$ and $\lambda_{\min}(\Sigma_X)$ denote the largest and smallest eigenvalues of the matrix Σ_X. We make use of these bounds in our experiments.

As in the paper [43], we also consider a preconditioned version of the method; more precisely, we first sample from $\Pi_g^* \propto e^{-g}$ where $g(\theta) = f(\Sigma_X^{-1/2}\theta)$, and then transform the obtained random samples $\theta_i \mapsto \Sigma_X^{1/2}\theta_i$ to obtain samples from Π^*. Sampling based on the preconditioned distribution improves the condition number of the problem. After the preconditioning, we have the bounds $L_g \leq 0.25n + \alpha$ and $m_g \geq \alpha$, so that the new condition number is now independent of the eigenvalues of Σ_X.

We randomly draw i.i.d. samples (x_i, y_i) as follows. Each vector $x_i \in \mathbb{R}^d$ is sampled i.i.d. Rademacher components, and then renormalized to Euclidean norm. given x_i, the

response y_i is drawn from the logistic model (2.24) with $\theta = \theta^* = \mathbf{1}_d = (1, \ldots, 1)^\top$. We fix $d = 2, n = 50$ and perform $N = 1000$ experiments. To sample from the posterior, we start with the initial distribution as $\mu_0 = \mathcal{N}(0, L^{-1}\mathbb{I}_d)$. As the first error metric, we measure the ℓ_1 distance between the true parameter θ^* and the sample mean $\hat{\theta}_k$ of the random samples obtained from simulating the Markov chains for k iterations:

$$e_k = \frac{1}{d}\|\hat{\theta}_k - \theta^*\|_1.$$

Figure 2.7 shows this error as a function of iteration number in logarithmic scale. Since there is always an approximation error caused by the prior distribution, ULA with large step-size ($\delta = 1.0$) can be used. However, our simulation shows that it is still slower than MALA. Also, the condition number κ has a significant effect on the mixing time of ULA and MRW. Their convergence in the preconditioned case is significantly better. Furthermore, the autocorrelation plots in Figure 2.8 and the plots in Figure 2.9 of the sample (across experiments) mean and 25% and 75% quantiles, with θ^* subtracted, as a function of iterations suggest a similar story: MALA converges faster than ULA and is less affected by conditioning of the problem.

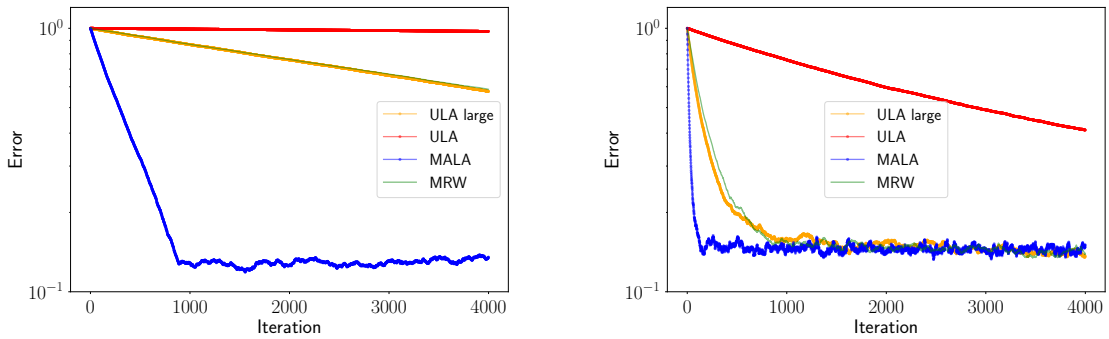


Figure 2.7. Mean error as a function of iteration number. (a) Without preconditioning. (b) With preconditioning.

2.4.4 Step size vs accept-reject rate

In this section, we provide a few simulations that highlight the effect of step size for MALA and MRW. Note that our bounds from Theorem 1 and 2 suggest a step size choice of order d^{-1} for both MALA and MRW, which in turn led to the mixing time bounds of Od. These choices of step sizes arise when we try to provide a worst-case control on the accept-reject step of these algorithms. In particular, these choices ensure that the Markov chains do not get stuck at a given state x, or equivalently, that the proposals at any given state are accepted with constant probability. If instead, one

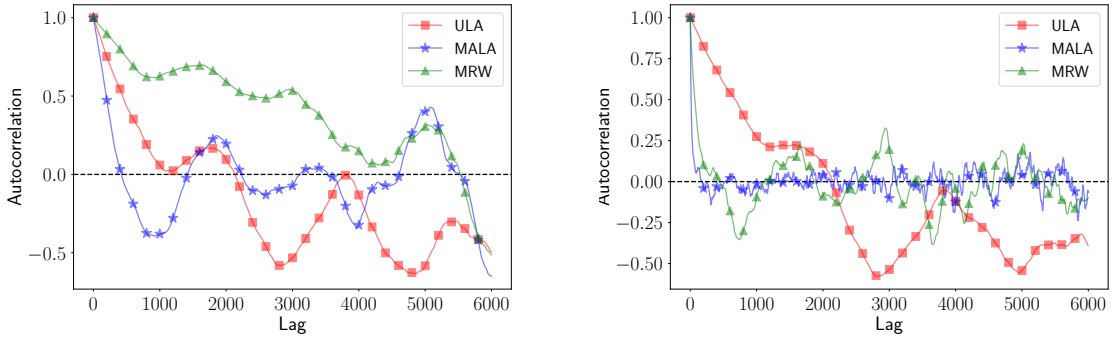


Figure 2.8. Autocorrelation function plot of the first coordinate of the estimate as a function lag. The burn-in time for the plot is set to 300 iterations. (a) Without preconditioning. (b) With preconditioning.

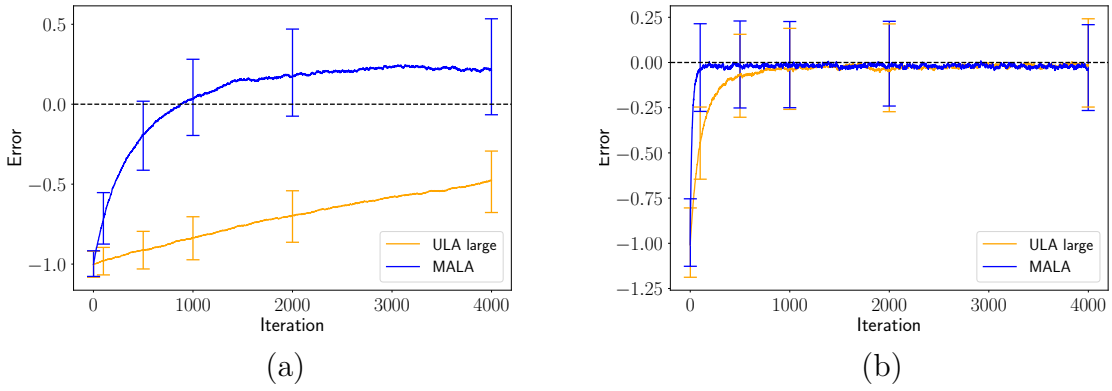
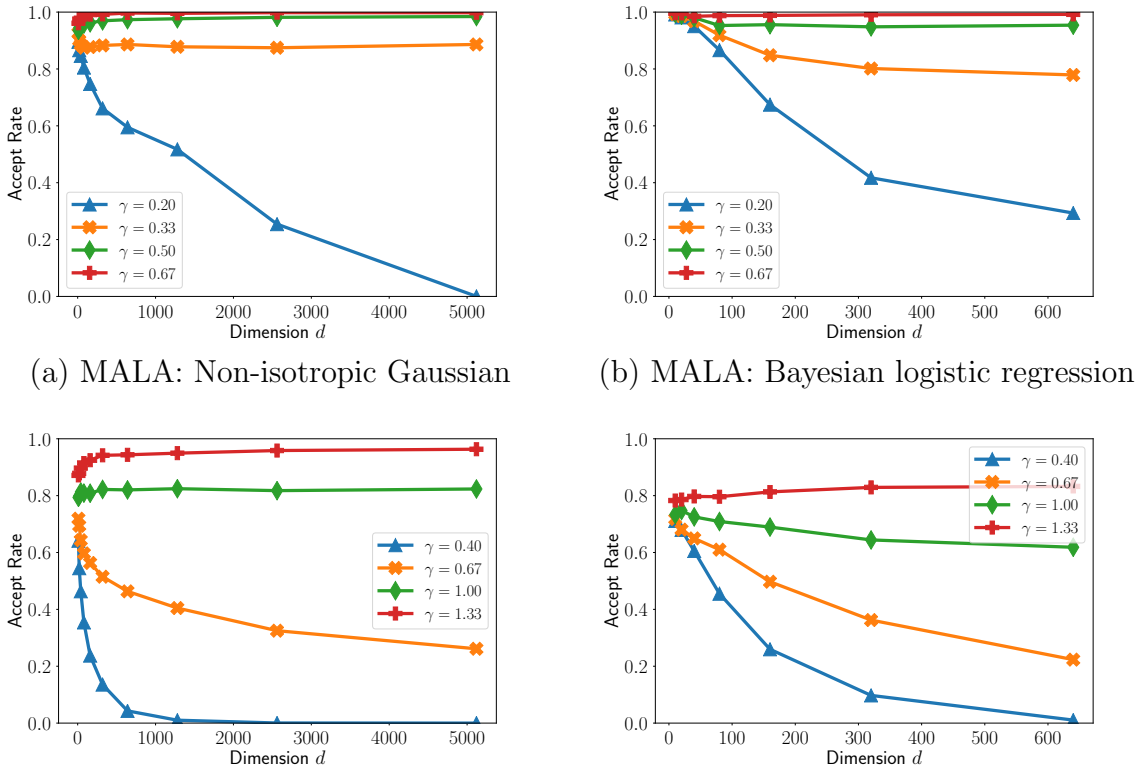


Figure 2.9. Mean and 25% and 75% quantiles, with θ^* subtracted, as a function of iteration number. (a) Without preconditioning. (b) With preconditioning.

chooses a very large step size, the (worst-case) probability of acceptance may decay exponentially with dimensions. Nonetheless, these worst case bounds may not hold, which would imply a faster mixing time for these chains if a larger step size were to be used.

To check the validity of larger step sizes, we repeated a few experiments discussed above, albeit with a larger step size. In particular, we simulated the random walks for a wide-range of step sizes $d^{-\gamma}$ for $\gamma \in \{0.2, 0.33, 0.5, 0.67\}$ for MALA, and, $\gamma \in \{0.4, 0.67, 1, 1.33\}$ for MRW. We ran these chains for two different cases: (a) Sampling from non-isotropic Gaussian density, discussed in Section 2.4.1, and, (b) Posterior sampling in Bayesian logistic regression, discussed in Section 2.4.3). In Figure 2.10, we plot the average acceptance probability for different step sizes $d^{-\gamma}$ as the dimension d

increases. These probabilities were computed as the average number of proposals accepted over 100 iterations after a manually tuned burn-in period, and further averaged across 50 independent runs.



(a) MALA: Non-isotropic Gaussian

(b) MALA: Bayesian logistic regression

(c) MRW: Non-isotropic Gaussian

(d) MRW: Bayesian logistic regression

Figure 2.10. Effect of large step size for accept-reject ratio for MALA and MRW. From panels (a) and (b), we see that for MALA the step size choice of $d^{-0.5}$ has a non-vanishing acceptance probability rate for both cases. On the other hand, panels (c) and (d) show that for MRW d^{-1} is a good choice for the step size.

We now remark on the observations from Figure 2.10. We see that for MALA the acceptance probability for the step size choice of $d^{-0.2}$ vanishes as d increases. Indeed, the choice of $d^{-0.5}$ appears to be a safe choice for both cases. In contrast, for MRW, we need a smaller step size. From panels (c) and (d), we see that d^{-1} appears to be the correct choice to ensure that the proposal are accepted with a constant probability when the dimension d is large.

Informally, if a step size choice of $d^{-\gamma}$ were to guarantee a non-vanishing acceptance probability for MALA or MRW, our proof techniques imply a mixing time bound of $O\left(d^{\gamma}\right)$. Combining this argument with the observations above, we suspect that the bounds for MALA from Theorem 1 may not be tight, while for MRW the bounds from Theorem 2 are very likely to be tight. Deriving a faster mixing time for MALA or

establishing that the current dimension dependency for MRW is tight, are interesting research directions and we leave further investigation of these questions for future work.

2.5 Proofs

We now turn to the proofs of our main results. In Section 2.5.1, we begin by introducing some background on conductance bounds, before stating three auxiliary lemmas that underlie the proofs of our main theorems. Taking these three lemmas as given, we then provide the proof of Theorem 1 in Section 2.5.2. Sections 2.5.3 through 2.5.5 are devoted to the proofs of our three key lemmas, and we conclude with the proof of Theorem 2 in Section 2.5.6.

2.5.1 Conductance bounds and auxiliary results

Our proofs exploit standard conductance-based arguments for controlling mixing times. Consider an ergodic Markov chain defined by a transition operator $\mathcal{T}$, and let Π^* be its stationary distribution. For each scalar $s \in (0, 1/2)$, we define the s-conductance

$$\Phi_s := \inf_{\Pi^*(A) \in (s, 1-s)} \frac{\int_A \mathcal{T}_u(A^c)\Pi^*(u)du}{\min\{\Pi^*(A) - s, \Pi^*(A^c) - s\}}. \tag{2.25}$$

In this formula, the notation $\mathcal{T}_u$ is shorthand for the distribution $\mathcal{T}(\delta_u)$ obtained by applying the transition operator to a dirac distribution concentrated on u. In words, the s-conductance measures how much probability mass flows across disjoint sets relative to their stationary mass. By a continuity argument, it can be seen that limiting conductance of the chain is equal to the limiting value of s-conductance—that is, $\Phi = \lim_{s \to 0} \Phi_s$.

For a reversible lazy Markov chain with ϖ-warm start, Lovász [95, 115] proved that

$$d_{\mathrm{TV}}\left(\mathcal{T}^k(\mu_0), \Pi^*\right) \leq \varpi s + \varpi\left(1 - \frac{\Phi_s^2}{2}\right)^k \leq \varpi s + \varpi e^{-k\Phi_s^2/2} \quad \text{for any } s \in \left(0, \tfrac{1}{2}\right). \tag{2.26}$$

In order to make effective use of this lower bound, we need to lower bound the s-conductance Φ_s, and then choose the parameter s so as to optimize the tradeoff between the two terms in the bound. We now state some auxiliary results that are useful.

We start with a result that shows that the probability mass of any strongly log concave distributions is concentrated in a Euclidean ball around the mode. For each $s \in (0, 1)$, we introduce the Euclidean ball

$$\mathcal{R}_s = \mathbb{B}\left(x^\star, r(s)\sqrt{\frac{d}{m}}\right) \tag{2.27}$$

where the function r was previously defined in equation (2.13a), and $x^\star := \arg\max_{x \in \mathbb{R}^d} \Pi^*(x)$ denotes the mode.

Lemma 1. *For any $s \in \left(0, \frac{1}{2}\right)$, we have $\Pi^*(\mathcal{R}_s) \geq 1 - s$.*

See Section 2.5.3 for the proof of this claim.

In order to establish the conductance bounds inside this ball, we first prove an extension of a result by Lovász [115]. It provides a lower bound on the flow of Markov chain with transition distribution $\mathcal{T}_x$ and strongly log concave target distributions Π^*.

Lemma 2. *Let Ω be a convex set such that $d_{TV}(\mathcal{T}_x, \mathcal{T}_y) \leq 1 - \rho$ whenever $x, y \in \Omega$ and $\|x - y\|_2 \leq \Delta$. Then for any measurable partition A_1 and A_2 of $\mathbb{R}^d$, we have*

$$\int_{A_1} \mathcal{T}_u(A_2)\Pi^*(u)du \geq \frac{\rho}{4} \min\left\{1, \frac{\log 2 \cdot \Delta \cdot \Pi^*(\Omega)^2 \cdot \sqrt{m}}{8}\right\} \min\left\{\Pi^*(A_1 \cap \Omega), \Pi^*(A_2 \cap \Omega)\right\}. \tag{2.28}$$

See Section 2.5.4 for the proof of this lemma.

We next introduce a few pieces of notations to state a MALA specific result. Define a function $\tilde{w} : (0,1) \times (0,1) \to \mathbb{R}_+$ as follows:

$$\tilde{w}(s, \varepsilon) := \min\left\{\frac{\sqrt{\varepsilon}}{8\sqrt{2}r(s)} \frac{\sqrt{m}}{L\sqrt{dL}}, \frac{\varepsilon}{64\alpha_\varepsilon} \frac{1}{Ld}, \frac{\varepsilon^{2/3}}{26(\alpha_\varepsilon r^2(s))^{1/3}} \frac{1}{L} \left(\frac{m}{Ld^2}\right)^{1/3}\right\}, \tag{2.29a}$$

$$\text{where} \quad \alpha_\varepsilon := 1 + 2\sqrt{\log(16/\varepsilon)} + 2\log(16/\varepsilon), \tag{2.29b}$$

and the function r was defined in equation (2.13a).

In the next lemma, we show two important properties for MALA: (1) the proposal distributions of MALA at two different points are close if the two points are close, and (2) the accept-reject step of MALA is well behaved inside the ball $\mathcal{R}_s$ provided the step size is chosen carefully. Note that for MALA, the proposal distribution of the chain at x is given by

$$\mathcal{P}_x^{\text{MALA}(\eta)} = \mathcal{N}(\mu_x, 2\eta\mathbb{I}_d), \quad \text{where} \quad \mu_x = x - \eta\nabla f(x). \tag{2.30}$$

We use $\mathcal{T}_x^{\text{MALA}(\eta)}$ to denote the transition distribution of MALA.

Lemma 3. *For any step size $\eta \in \left(0, \frac{2}{L}\right]$, the MALA proposal distribution satisfies the bound*

$$\sup_{\substack{x,y \in \mathbb{R}^d \\ x \neq y}} \frac{d_{TV}\left(\mathcal{P}_x^{MALA(\eta)}, \mathcal{P}_y^{MALA(\eta)}\right)}{\|x - y\|_2} \leq \frac{1}{\sqrt{2\eta}}. \tag{2.31a}$$

Moreover, given scalars $s \in (0, 1/2)$ and $\varepsilon \in (0,1)$, then the MALA proposal distribution for any step size $\eta \in \left(0, \tilde{w}(s,\varepsilon)\right]$ satisfies the bound

$$\sup_{x \in \mathcal{R}_s} d_{TV}\left(\mathcal{P}_x^{MALA(\eta)}, \mathcal{T}_x^{MALA(\eta)}\right) \leq \frac{\varepsilon}{8}, \tag{2.31b}$$

where the truncated ball $\mathcal{R}_s$ was defined in equation (2.27).

See Section 2.5.5 for the proof.

With these results in hand, we now prove the mixing time bound for MALA.

2.5.2 Proof of Theorem 1

At a high level, the proof involves three key steps. Our first step is to use Lemma 3 to establish that for an appropriate choice of step size, the MALA update has nice properties inside a high probability region given by Lemma 1. The second step is to apply Lemma 2 so as to obtain a lower bound on the s-conductance Φ_s of the MALA update. Finally, by making an appropriate choice of parameter s, we establish the claimed convergence rate.

So as to simplify notation, we drop the superscripts $\text{MALA}(\eta)$ from our notation—that is, we use $\mathcal{T}_x$ and $\mathcal{P}_x$, respectively, to denote the transition and proposal distributions at x for MALA, each with step size η. By applying the triangle inequality, we obtain the upper bound

$$d_{\text{TV}}(\mathcal{T}_x, \mathcal{T}_y) \leq d_{\text{TV}}(\mathcal{P}_x, \mathcal{T}_x) + d_{\text{TV}}(\mathcal{P}_x, \mathcal{P}_y) + d_{\text{TV}}(\mathcal{P}_y, \mathcal{T}_y). \tag{2.32}$$

Now applying claim (2.31a) from Lemma 3 guarantees that

$$d_{\text{TV}}(\mathcal{P}_x, \mathcal{P}_y) \leq \varepsilon/\sqrt{2} \qquad \text{for all } x, y \in \mathbb{R}^d \text{ such that } \|x - y\|_2 \leq \varepsilon\sqrt{\eta}.$$

Furthermore, for any $\eta \leq \tilde{w}(s, \varepsilon)$, the bound (2.31b) from Lemma 3 implies that $d_{\text{TV}}(\mathcal{P}_x, \mathcal{T}_x) \leq \varepsilon/8$ for any $x \in \mathcal{R}_s$. Plugging in these bounds in the inequality (2.32), we find that

$$d_{\text{TV}}(\mathcal{T}_x, \mathcal{T}_y) \leq 1 - (1 - \varepsilon) \quad \forall\, x, y \in \mathcal{R}_s \text{ such that } \|x - y\|_2 \leq \varepsilon\sqrt{\eta}.$$

Thus, the transition distribution $\mathcal{T}_x$ satisfies the assumptions of Lemma 2 for

$$\Omega = \mathcal{R}_s, \quad \rho = (1 - \varepsilon) \quad \text{and} \quad \Delta = \varepsilon\sqrt{\eta}. \tag{2.33}$$

We now derive a lower bound on the s-conductance of MALA. Choosing a measurable set A such that $\Pi^*(A) > s$ and substituting the terms from equation (2.33) in the inequality (2.28), we find that

$$\begin{aligned}\int_A \mathcal{T}_u(A^c)\Pi^*(u)du &\geq \frac{(1-\varepsilon)}{4} \min\left\{1, \frac{\log 2 \cdot \varepsilon\sqrt{\eta} \cdot \Pi^*(\mathcal{R}_s)^2 \cdot \sqrt{m}}{8}\right\} \cdot \\ &\qquad \cdot \min\left\{\Pi^*(A \cap \mathcal{R}_s), \Pi^*(A^c \cap \mathcal{R}_s)\right\} \\ &\overset{(i)}{\geq} \frac{(1-\varepsilon)\varepsilon\sqrt{\eta} \cdot \Pi^*(\mathcal{R}_s)^2 \cdot \sqrt{m}}{64} \min\left\{\Pi^*(A) - s, \Pi^*(A^c) - s\right\}.\end{aligned}$$

In this argument, inequality (i) follows from the facts that $\log 2 \geq 1/2$ and $\Pi^*(A), \Pi^*(A^c) > s$. Moreover, we have applied Lemma 1 to find that $\Pi^*(\mathcal{R}_s) \geq 1 - s$ and hence

$$\Pi^*(\mathcal{X} \cap \mathcal{R}_s) = \Pi^*(\mathcal{X}) - \Pi^*(\mathcal{X} \cap \mathcal{R}_s^c) \geq \Pi^*(\mathcal{X}) - s \quad \text{for } \mathcal{X} \in \{A, A^c\}.$$

We have also assumed that the second argument of the minimum is less than 1. Applying the definition (2.25) of Φ_s for MALA, we find that

$$\Phi_s^{\text{MALA}(\eta)} \geq \frac{(1-\varepsilon)\varepsilon \cdot \Pi^*(\mathcal{R}_s)^2 \cdot \sqrt{\eta m}}{64}, \qquad \text{for any } \eta \leq \tilde{w}(s, \varepsilon). \tag{2.34}$$

By making a suitable choice of s, we can now complete the proof. Using Lemma 1, we have that $\Pi^*(\mathcal{R}_{\epsilon/2}) \geq 1 - \epsilon/2 \geq 1/2$ for any $\epsilon \in (0,1)$. Applying the definition (2.29b) of α_ε, we obtain that $\alpha_{1/2} \leq 12$. Using this fact and the definitions (2.13b) and (2.29a) for the functions $w(\cdot)$ and $\tilde{w}(\cdot,\cdot)$, it is straightforward to verify that $cw(\epsilon/(2\varpi)) \leq \tilde{w}(\epsilon/(2\varpi), 1/2)$, for an appropriate choice of universal constant c. Substituting in $s = \epsilon/(2\varpi)$, $\varepsilon = 1/2$, and $\eta = cw(\epsilon/(2\varpi))$, and also making use of the lower bound $\Pi^*(\mathcal{R}_{\epsilon/2\varpi}) \geq 1/2$ in the bound (2.34), we find that $\Phi^{\mathrm{MALA}(\eta)}_{\epsilon/2\varpi} \geq c'\sqrt{m\eta}$ for some universal constant c'. Using the convergence rate (2.26), we obtain that

$$d_{\mathrm{TV}}\left(\mathcal{T}^k_{\mathrm{MALA}(\eta)}(\mu_0), \Pi^*\right) \leq \varpi\frac{\epsilon}{2\varpi} + \varpi e^{-km\eta/c'} \leq \epsilon \quad \text{for all} \quad k \geq \frac{c'}{m\eta} \cdot \log\left(\frac{2\varpi}{\epsilon}\right), \tag{2.35}$$

for a suitably large constant c'. Substituting the expression (2.13b) for $\eta = cw(\epsilon/(2\varpi))$, yields the claimed bound on mixing time.

2.5.3 Proof of Lemma 1

The proof consists of two main steps. First, we establish that the distribution Π^* is sub-Gaussian, which then guarantees concentration around the mean. Second, we show that the mean and the mode of the distribution Π^* are not far apart. Combining these two claims yields a high probability region around the mode $x^\star$.

Let x denote the random variable with distribution Π^* and mean $\bar{x} = \mathbb{E}_{x\sim\Pi^*}[x]$. We claim that $x - \bar{x}$ is a sub-Gaussian random vector with parameter $1/\sqrt{m}$, meaning that

$$\mathbb{E}_x\left[e^{u^\top(x-\bar{x})}\right] \leq e^{\|u\|_2^2/(2m)} \quad \text{for any vector } u \in \mathbb{R}^d.$$

In order to prove this claim, we make use of a result due to Hargé (Theorem 1.1 [76]), which we restate here. Let $y \sim \mathcal{N}(\mu, \Sigma)$ with density e and x be a random variable with density function $q \cdot e$ where q is a log-concave function. Then for any convex function $g : \mathbb{R}^d \mapsto \mathbb{R}$ we have

$$\mathbb{E}_x\left[g(x - \mathbb{E}[x])\right] \leq \mathbb{E}_y\left[g(y - \mathbb{E}[y])\right]. \tag{2.36}$$

f is m-strongly convex, we have that $x \mapsto f(x) - \frac{m}{2}\|x - x^\star\|_2^2$ is a convex function. Thus we can express the density π^* as the product of a log concave function and the density of a random variable with distribution $\mathcal{N}(x^\star, \mathbb{I}_d/m)$. Letting $y \sim \mathcal{N}(x^\star, \mathbb{I}_d/m)$ and noting that $g(z) := e^{u^\top z}$ is a convex function for each fixed vector u, applying the Hargé bound (2.36) yields

$$\mathbb{E}_x\left[e^{u^\top(x-\bar{x})}\right] \leq \mathbb{E}_y\left[e^{u^\top(y-x^*)}\right] \overset{(i)}{\leq} e^{\|u\|_2^2/2m}.$$

Here inequality (i) follows from the fact that the random vector $y - x^\star$ is sub-Gaussian with parameter $1/\sqrt{m}$.

Using the standard tail bounds for quadratic forms for sub-Gaussian random vectors (e.g., Theorem 1 [82]), we find that

$$\mathbb{P}_{x\sim\Pi^*}\left[\|x-\bar{x}\|_2^2 > \frac{d}{m}\left(1+2\sqrt{\frac{t}{d}}+2\frac{t}{d}\right)\right] \leq e^{-t}. \tag{2.37}$$

Define $\mathcal{B}_1 := \mathbb{B}\left(\bar{x}, \sqrt{\frac{d}{m}}\cdot\tilde{r}(s)\right)$ where $\tilde{r}(s) = 1+2\max\left\{\left(\frac{\log(1/s)}{d}\right)^{0.25}, \sqrt{\frac{\log(1/s)}{d}}\right\}$. Observe that $\tilde{r}(s)^2 \geq 1+2\sqrt{\frac{\log(1/s)}{d}}+2\frac{\log(1/s)}{d}$ and consequently the bound (2.37) implies that $\Pi^*(\mathcal{B}_1) = \mathbb{P}_{x\sim\Pi^*}[x\in\mathcal{B}_1] \geq 1-s$. Now applying triangle inequality, we obtain that

$$\mathcal{B}_1 \subseteq \mathbb{B}\left(x^\star, \|\bar{x}-x^\star\|_2 + \sqrt{\frac{d}{m}}\cdot\tilde{r}(s)\right) =: \mathcal{B}_2$$

From Theorem 1 by Durmus et al. [55], we have that $\mathbb{E}_{x\sim\Pi^*}\|x-x^\star\|_2^2 \leq d/m$. Using Jensen inequality twice, we find that

$$\|\bar{x}-x^\star\|_2 = \|\mathbb{E}_{x\sim\Pi^*}[x]-x^\star\|_2 \leq \mathbb{E}_{x\sim\Pi^*}\|x-x^\star\|_2 \leq \sqrt{\mathbb{E}_{x\sim\Pi^*}\|x-x^\star\|_2^2} \leq \sqrt{\frac{d}{m}}. \tag{2.38}$$

Noting the relation $r(s) = 1+\tilde{r}(s)$, we thus obtain that $\|\bar{x}-x^\star\|_2+\sqrt{\frac{d}{m}}\cdot\tilde{r}(s) \leq r(s)\sqrt{\frac{d}{m}}$ and consequently $\mathcal{B}_1 \subseteq \mathcal{B}_2 \subseteq \mathcal{R}_s$. As a result, we obtain $\Pi^*(\mathcal{R}_s) \geq \Pi^*(\mathcal{B}_1) \geq 1-s$ as claimed.

2.5.4 Proof of Lemma 2

The proof of this lemma is based on the following isoperimetric inequality for log-concave distributions. Let $\mathbb{R}^d = S_1 \cup S_2 \cup S_3$ be a partition. Let $y \sim \mathcal{N}(0, \sigma^2\mathbb{I}_d)$ with density e and let Π^* be a distribution with a density given by $q\cdot e$ where q is a log-concave function. Then Cousins and Vempala (Theorem 4.4 [40]) proved that

$$\Pi^*(S_3) \geq \frac{\log 2\cdot d(S_1,S_2)}{\sigma}\Pi^*(S_1)\Pi^*(S_2) \tag{2.39}$$

where $d(S_1,S_2) := \inf\left\{\|x-y\|_2 \,\middle|\, x\in S_1, y\in S_2\right\}$.

We invoke this result for the truncated distribution π^*_Ω with the density π^*_Ω defined as

$$\pi^*_\Omega(x) := \frac{1}{\int_\Omega \pi^*(y)dy}\pi^*(x)\mathbf{1}_\Omega(x) = \frac{1}{\int_\Omega e^{-f(y)}dy}e^{-f(x)}\mathbf{1}_\Omega(x), \tag{2.40}$$

where $\mathbf{1}_\Omega(\cdot)$ denotes the indicator function for the set Ω, i.e., we have $\mathbf{1}_\Omega(x) = 1$ if $x \in \Omega$, and 0 otherwise. Let $x^\star = \arg\max \Pi^*(x) = \arg\min f(x)$. Observe that m-strong-convexity of f implies that $x \mapsto f(x) - \frac{m}{2}\|x - x^\star\|_2^2$ is a convex function. Noting that the function $\mathbf{1}_\Omega(\cdot)$ is log-concave and that log-concavity is closed under multiplication, we conclude that π_Ω^* can be expressed as a product of log-concave function and density of the Gaussian distribution $\mathcal{N}\left(x^\star, \frac{1}{m}\mathbb{I}_d\right)$. Consequently, we can apply the result (2.39) with Π^* replaced by π_Ω^* and $\sigma = 1/\sqrt{m}$.

We now prove the claim of the lemma. Define the sets

$$A_1' := \left\{u \in A_1 \cap \Omega \;\middle|\; \mathcal{T}_u(A_2) < \frac{\rho}{2}\right\}, \quad A_2' := \left\{v \in A_2 \cap \Omega \;\middle|\; \mathcal{T}_v(A_1) < \frac{\rho}{2}\right\}, \tag{2.41}$$

along with the complement $A_3' := \Omega \backslash (A_1' \cup A_2')$. See Figure 2.11 for an illustration. Based on these three sets, we split our proof of the claim (2.28) into two distinct cases:

- Case 1: $\Pi^*(A_1') \leq \Pi^*(A_1 \cap \Omega)/2$ or $\Pi^*(A_2') \leq \Pi^*(A_2 \cap \Omega)/2$.
- Case 2: $\Pi^*(A_i') \geq \Pi^*(A_i \cap \Omega)/2$ for $i = 1, 2$.

Note that these cases are mutually exclusive, and cover all possibilities.

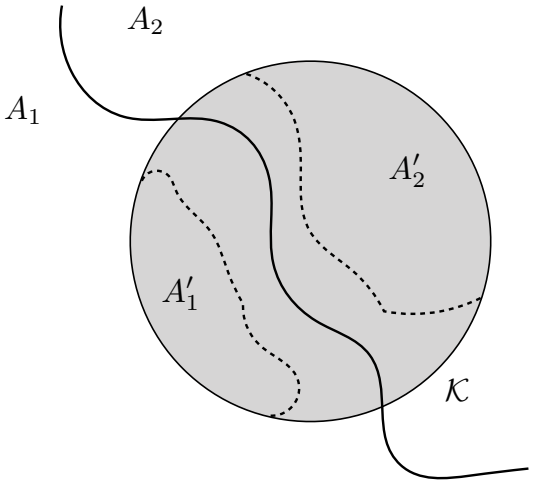


Figure 2.11. The sets A_1 and A_2 form a partition of $\mathbb{R}^d$, and we use Ω to denote a compact convex subset. The sets A_1' and A_2' are defined in equation (2.41).

Case 1 We have $\Pi^*(A_1 \cap \Omega \backslash A_1') \geq \Pi^*(A_1 \cap \Omega)/2$, then

$$\int_{A_1} \mathcal{T}_u(A_2)\Pi^*(u)du \overset{(i)}{\geq} \int_{A_1 \cap \Omega \backslash A_1'} \mathcal{T}_u(A_2)\Pi^*(u)du \overset{(ii)}{\geq} \frac{\rho}{2}\Pi^*(A_1 \cap \Omega \backslash A_1') \overset{(iii)}{\geq} \frac{\rho}{4}\Pi^*(A_1 \cap \Omega),$$

which implies the claim (2.28). In the above sequence of inequalities, step (i) is trivially true; step (ii) from the definition (2.41) of the set A_1', and step (iii) from the assumption for this case.

A similar argument with the roles of A_1 and A_2 switched, establishes the claim when $\Pi^*(A_2') \leq \Pi^*(A_2 \cap \Omega)/2$.

Case 2 We have $\Pi^*(A_i') \geq \Pi^*(A_i \cap \Omega)/2$ for both $i = 1$ and 2. For any $u \in A_1'$ and $v \in A_2'$, we have that

$$d_{\mathrm{TV}}(\mathcal{T}_u, \mathcal{T}_v) \geq \mathcal{T}_u(A_1) - \mathcal{T}_v(A_1) \overset{(i)}{=} 1 - \mathcal{T}_u(A_2) - \mathcal{T}_v(A_1) > 1 - \rho,$$

where step (i) follows from the fact that $A_1 = \mathbb{R}^d \backslash A_2$ and thereby $\mathcal{T}_u(A_1) = 1 - \mathcal{T}_u(A_2)$. Since $u, v \in \Omega$, the assumption of the lemma implies that $\|u - v\|_2 \geq \Delta$ and consequently

$$d(A_1', A_2') \geq \Delta. \tag{2.42}$$

We claim that

$$\int_{A_1} \mathcal{T}_u(A_2)\Pi^*(u)du = \int_{A_2} \mathcal{T}_v(A_1)\Pi^*(v)dv \tag{2.43}$$

We provide the proof of this claim at the end. Assuming this claim as given, we now complete the proof. Using equation (2.43), we have

$$\begin{aligned}
\int_{A_1} \mathcal{T}_u(A_2)\Pi^*(u)du &= \frac{1}{2}\left(\int_{A_1} \mathcal{T}_u(A_2)\Pi^*(u)du + \int_{A_2} \mathcal{T}_v(A_1)\Pi^*(v)dv\right) \\
&\geq \frac{1}{4}\left(\int_{A_1 \cap \Omega \backslash A_1'} \mathcal{T}_u(A_2)\Pi^*(u)du + \int_{A_2 \cap \Omega \backslash A_2'} \mathcal{T}_v(A_1)\Pi^*(v)dv\right) \\
&\overset{(i)}{\geq} \frac{\rho}{8}\Pi^*(\Omega \backslash (A_1' \cup A_2')),
\end{aligned} \tag{2.44}$$

where step (i) follows from the definition (2.41) of the set $A_3' = \Omega \backslash (A_1' \cup A_2')$. Further, we have

$$\begin{aligned}
\Pi^*(\Omega \backslash (A_1' \cup A_2')) &\overset{(i)}{=} \Pi^*(\Omega) \cdot \pi_\Omega^*(\Omega \backslash A_1' \backslash A_2') \\
&\overset{(ii)}{\geq} \Pi^*(\Omega) \cdot \frac{\log 2 \cdot d(A_1', A_2')}{1/\sqrt{m}} \cdot \pi_\Omega^*(A_1') \cdot \pi_\Omega^*(A_2') \\
&\overset{(iii)}{\geq} \Pi^*(\Omega) \cdot \log 2 \cdot d(A_1', A_2') \cdot \sqrt{m} \cdot \Pi^*(A_1') \cdot \Pi^*(A_2') \\
&\overset{(iv)}{\geq} \Pi^*(\Omega) \cdot \log 2 \cdot \Delta \cdot \sqrt{m} \cdot \frac{1}{4} \cdot \Pi^*(A_1 \cap \Omega) \cdot \Pi^*(A_2 \cap \Omega). \qquad (2.45)
\end{aligned}$$

where step (i) follows from the definition (2.40) of the truncated distribution π_Ω^*, step (ii) follows from applying the isoperimetry (2.39) for the distribution π_Ω^* with $\sigma = 1/\sqrt{m}$, step (iii) from the definition of π_Ω^* and step (iv) from inequality (2.42) and the assumption for this case. Let $\alpha := \Pi^*(A_1 \cap \Omega)/\Pi^*(\Omega)$. Note that $\alpha \in [0, 1]$ and $\Pi^*(A_2 \cap \Omega)/\Pi^*(\Omega) = 1 - \alpha$. We have

$$\begin{aligned}
\Pi^*(A_1 \cap \Omega) \cdot \Pi^*(A_2 \cap \Omega) &= \Pi^*(\Omega)^2 \cdot \alpha(1 - \alpha) \\
&\geq \Pi^*(\Omega)^2 \cdot \frac{1}{2} \min\{\alpha, 1 - \alpha\} \\
&= \Pi^*(\Omega) \cdot \frac{1}{2} \min\{\Pi^*(A_1 \cap \Omega), \Pi^*(A_2 \cap \Omega)\} \qquad (2.46)
\end{aligned}$$

Putting the inequalities (2.44), (2.45) and (2.46) together, establishes the claim (2.28) of the lemma for this case.

We now prove our earlier claim (2.43). Note that it suffices to prove that

$$\int_{A_1} \mathcal{T}_u(A_2) \Pi^*(u) du = \int_{A_2} \mathcal{T}_v(A_1) \Pi^*(v) dv.$$

We have

$$\begin{aligned}
\int_{A_2} \mathcal{T}_u(A_1) \Pi^*(u) du &\overset{(i)}{=} \int_{\mathbb{R}^d} \mathcal{T}_u(A_1) \Pi^*(u) du - \int_{A_1} \mathcal{T}_u(A_1) \Pi^*(u) du \\
&\overset{(ii)}{=} \Pi^*(A_1) - \int_{A_1} \mathcal{T}_u(A_1) \Pi^*(u) du \\
&= \int_{A_1} \Pi^*(u) du - \int_{A_1} \mathcal{T}_u(A_1) \Pi^*(u) du \\
&\overset{(iii)}{=} \int_{A_1} \mathcal{T}_u(A_2) \Pi^*(u) du,
\end{aligned}$$

where steps (i) and (iii) (respectively) follow from the fact that $A_1 = \mathbb{R}^d \backslash A_2$ and the consequent fact that $1 - \mathcal{T}_u(A_1) = \mathcal{T}_u(A_2)$, and step (ii) follows from the fact that Π^* is the stationary density for the transition distribution $\mathcal{T}_x$ and thereby $\int_{\mathbb{R}^d} \mathcal{T}_u(A_1) \Pi^*(u) du = \Pi^*(A_1)$.

2.5.5 Proof of Lemma 3

We prove each claim of the lemma separately. To simplify notation, we drop the superscript from our notations of distributions $\mathcal{T}_x^{\mathrm{MALA}(\eta)}$ and $\mathcal{P}_x^{\mathrm{MALA}(\eta)}$.

Proof of claim (2.31a)

In order to bound the total variation distance $d_{\mathrm{TV}}(\mathcal{P}_x, \mathcal{P}_y)$, we apply Pinsker's inequality [41], which guarantees that $d_{\mathrm{TV}}(\mathcal{P}_x, \mathcal{P}_y) \leq \sqrt{2\mathrm{KL}(\mathcal{P}_x \parallel \mathcal{P}_y)}$. Given multivariate normal distributions $\mathcal{G}_1 = \mathcal{N}(\mu_1, \Sigma)$ and $\mathcal{G}_2 = \mathcal{N}(\mu_2, \Sigma)$, the Kullback-Leibler divergence between the two is given by

$$\mathrm{KL}(\mathcal{G}_1 \parallel \mathcal{G}_2) = \frac{1}{2}(\mu_1 - \mu_2)^\top \Sigma^{-1}(\mu_1 - \mu_2). \tag{2.47}$$

Substituting $\mathcal{G}_1 = \mathcal{P}_x$ and $\mathcal{G}_2 = \mathcal{P}_y$ into the above expression and applying Pinsker's inequality, we find that

$$\begin{aligned} d_{\mathrm{TV}}(\mathcal{P}_x, \mathcal{P}_y) \leq \sqrt{2\mathrm{KL}(\mathcal{P}_x \parallel \mathcal{P}_y)} &= \frac{\|\mu_x - \mu_y\|_2}{\sqrt{2\eta}} \\ &\overset{(i)}{=} \frac{\|(x - \eta\nabla f(x)) - (y - \eta\nabla f(y))\|_2}{\sqrt{2\eta}}, \end{aligned}$$

where step (i) follows from the definition (2.30) of the mean μ_x. Consequently, in order to establish the claim (2.31a), it suffices to show that

$$\|(x - \eta\nabla f(x)) - (y - \eta\nabla f(y))\|_2 \leq \|x - y\|_2.$$

Recalling that $|||B|||_2$ denotes the ℓ_2-operator norm of a matrix B (equal to the maximum singular value), we have

$$\begin{aligned} \|(x - \eta\nabla f(x)) - (y - \eta\nabla f(y))\|_2 &= \left\| \int_0^1 \left[\mathbb{I} - \eta\nabla^2 f(x + t(x-y))\right](x-y)dt \right\|_2 \\ &\leq \int_0^1 \left\| \left[\mathbb{I} - \eta\nabla^2 f(x + t(x-y))\right](x-y) \right\|_2 dt \\ &\overset{(i)}{\leq} \sup_{z \in \mathbb{R}^d} |||\mathbb{I}_d - \eta\nabla^2 f(z)|||_2 \ \|x - y\|_2, \end{aligned}$$

where step (i) follows from the definition of the operator norm. m-strongly convexity and L-smoothness guarantee that the Hessian is sandwiched as $m\mathbb{I}_d \preceq \nabla^2 f(z) \preceq L\mathbb{I}_d$ for all $z \in \mathbb{R}^d$, where $\mathbb{I}_d$ denotes the d-dimensional identity matrix. From this Hessian sandwich, it follows that

$$|||\mathbb{I}_d - \eta\nabla^2 f(x)|||_2 = \max\{|1 - \eta L|, |1 - \eta m|\} \ < \ 1.$$

Putting together the pieces yields the claim.

Proof of claim (2.31b)

Let $\mathcal{P}_1$ be a distribution admitting a density ρ_1 on $\mathbb{R}^d$, and let $\mathcal{P}_2$ be a distribution which has an atom at x and admitting a density ρ_2 on $\mathbb{R}^d \setminus \{x\}$. The total variation distance between the distributions $\mathcal{P}_1$ and $\mathcal{P}_2$ is given by

$$d_{\mathrm{TV}}(\mathcal{P}_1, \mathcal{P}_2) = \frac{1}{2}\left(\mathcal{P}_2(\{x\}) + \int_{\mathbb{R}^d} |\rho_1(z) - \rho_2(z)|\, dz\right). \tag{2.48}$$

The accept-reject step for MALA implies that

$$\mathcal{T}_x(\{x\}) = 1 - \int_{\mathbb{R}^d} \min\left\{1, \frac{\Pi^*(z)\cdot\rho_z(x)}{\Pi^*(x)\cdot\rho_x(z)}\right\}\rho_x(z)dz, \tag{2.49}$$

where p_x denotes the density corresponding to the proposal distribution $\mathcal{P}_x = \mathcal{N}(x - \eta\nabla f(x), 2\eta\mathbb{I}_d)$. From this fact and the formula (2.48), we find that

$$\begin{aligned} d_{\mathrm{TV}}(\mathcal{P}_x, \mathcal{T}_x) &= \frac{1}{2}\left(\mathcal{T}_x(\{x\}) + \int_{\mathbb{R}^d}\rho_x(z)dz - \int_{\mathbb{R}^d}\min\left\{1, \frac{\Pi^*(z)\cdot\rho_z(x)}{\Pi^*(x)\cdot\rho_x(z)}\right\}\rho_x(z)dz\right) \\ &= \frac{1}{2}\left(2 - 2\int_{\mathbb{R}^d}\min\left\{1, \frac{\Pi^*(z)\cdot\rho_z(x)}{\Pi^*(x)\cdot\rho_x(z)}\right\}\rho_x(z)dz\right) \\ &= 1 - \mathbb{E}_{z\sim\mathcal{P}_x}\left[\min\left\{1, \frac{\Pi^*(z)\cdot\rho_z(x)}{\Pi^*(x)\cdot\rho_x(z)}\right\}\right]. \end{aligned} \tag{2.50}$$

By applying Markov's inequality, we obtain

$$\mathbb{E}_{z\sim\mathcal{P}_x}\left[\min\left\{1, \frac{\Pi^*(z)\cdot\rho_z(x)}{\Pi^*(x)\cdot\rho_x(z)}\right\}\right] \geq \alpha\,\mathbb{P}\left[\frac{\Pi^*(z)\cdot\rho_z(x)}{\Pi^*(x)\cdot\rho_x(z)} \geq \alpha\right] \quad \text{for all } \alpha \in (0,1]. \tag{2.51}$$

We now derive a high probability lower bound for the ratio $[\Pi^*(z)\rho_z(x)] \,/\, [\Pi^*(x)\rho_x(z)]$. Noting that $\Pi^*(x) \propto \exp(-f(x))$ and $\rho_x(z) \propto \exp\left(-\|x - \eta\nabla f(x) - z\|_2^2/(4\eta)\right)$, we have

$$\begin{aligned} \frac{\Pi^*(z)\cdot\rho_z(x)}{\Pi^*(x)\cdot\rho_x(z)} &= \frac{\exp\left(-f(z) - \frac{\|x - z + \eta\nabla f(z)\|_2^2}{4\eta}\right)}{\exp\left(-f(x) - \frac{\|z - x + \eta\nabla f(x)\|_2^2}{4\eta}\right)} \\ &= \exp\left(\frac{4\eta(f(x) - f(z)) + \|z - x + \eta\nabla f(x)\|_2^2 - \|x - z + \eta\nabla f(z)\|_2^2}{4\eta}\right). \end{aligned} \tag{2.52}$$

Keeping track of the numerator of this exponent, we find that

$$\begin{aligned}
&4\eta(f(x)-f(z))+\|z-x+\eta\nabla f(x)\|_2^2-\|x-z+\eta\nabla f(z)\|_2^2\\
&\quad=4\eta(f(x)-f(z))+\|z-x\|_2^2+\|\eta\nabla f(x)\|_2^2+2\eta(z-x)^\top\nabla f(x)\\
&\qquad-\|x-z\|_2^2-\|\eta\nabla f(z)\|_2^2-2\eta(x-z)^\top\nabla f(z)\\
&\quad=2\eta\underbrace{\left(f(x)-f(z)-(x-z)^\top\nabla f(x)\right)}_{M_1}+2\eta\underbrace{\left(f(x)-f(z)-(x-z)^\top\nabla f(z)\right)}_{M_2}\\
&\qquad+\eta^2\underbrace{\left(\|\nabla f(x)\|_2^2-\|\nabla f(z)\|_2^2\right)}_{M_3}.
\end{aligned}\tag{2.53}$$

Now we provide lower bounds for the terms M_i, $i=1,2,3$ defined in the above display. Since f is strongly convex and smooth, yielding

$$M_1\geq-\frac{L}{2}\|x-z\|_2^2,\quad\text{and}\quad M_2\geq\frac{m}{2}\|x-z\|_2^2.\tag{2.54}$$

In order to lower bound M_3, we observe that

$$\begin{aligned}
M_3=\|\nabla f(x)\|_2^2-\|\nabla f(z)\|_2^2&=\langle\nabla f(x)+\nabla f(z),\,\nabla f(x)-\nabla f(z)\rangle\\
&\overset{(i)}{\geq}-\|\nabla f(x)+\nabla f(z)\|_2\,\|\nabla f(x)-\nabla f(z)\|_2\\
&\overset{(ii)}{\geq}-\left(2\|\nabla f(x)\|_2+L\|x-z\|_2\right)L\|x-z\|_2,
\end{aligned}\tag{2.55}$$

where step (i) follows from the Cauchy-Schwarz's inequality and step (ii) from the triangle inequality and L-smoothness of the function f.

Combining the bounds (2.54) and (2.55) with equations (2.53) and (2.52), we have established that

$$\frac{\Pi^*(z)\cdot\rho_z(x)}{\Pi^*(x)\cdot\rho_x(z)}\geq\exp\left(\underbrace{-\frac{1}{4}(L-m)\|x-z\|_2^2-\frac{\eta}{4}\left(2L\|x-z\|_2\|\nabla f(x)\|_2+L^2\|x-z\|_2^2\right)}_{=:T}\right).\tag{2.56}$$

Now to provide a high probability lower bound for the term T, we make use of the standard chi-squared tail bounds and the following relation between x and z:

$$z\overset{(d)}{=}x-\eta\nabla f(x)+\sqrt{2\eta}\xi,$$

where $\xi\sim\mathcal{N}(0,\mathbb{I}_d)$ and $\overset{(d)}{=}$ denotes equality in distribution. We have

$$\|x-z\|_2=\|\eta\nabla f(x)+\xi\|_2\leq\eta\|\nabla f(x)\|_2+\sqrt{2\eta}\|\xi\|_2,$$

which also implies

$$\|x - z\|_2^2 \leq 2\eta^2 \|\nabla f(x)\|_2^2 + 4\eta \|\xi\|_2^2 .$$

Using these two inequalities, we find that

$$T \geq -\frac{(L-m)\eta^2}{2} \|\nabla f(x)\|_2^2 - (L-m)\eta \|\xi\|_2^2 - \frac{L\eta^2}{2} \|\nabla f(x)\|_2^2 \\ - \frac{L\eta\sqrt{\eta}}{\sqrt{2}} \|\nabla f(x)\|_2 \|\xi\|_2 - \frac{L^2\eta^3}{2} \|\nabla f(x)\|_2^2 - L^2\eta^2 \|\xi\|_2^2 .$$

Simplifying and using the fact that $L\eta \leq 1$, we obtain that

$$T \geq -2\left(L\eta^2 \|\nabla f(x)\|_2^2 + L\eta \|\xi\|_2^2 + L\eta\sqrt{\eta} \|\nabla f(x)\|_2 \|\xi\|_2\right).$$

Since $x \in \mathcal{R}_s$, we have

$$\|\nabla f(x)\|_2 = \|\nabla f(x) - \nabla f(x^\star)\|_2 \overset{(i)}{\leq} L \|x - x^\star\|_2 \leq L\sqrt{\frac{d}{m}} r(s) =: \mathcal{D}_s, \tag{2.57}$$

where inequality (i) follows from L-smoothness. Thus, we have shown that

$$T \geq -2\left(L\eta^2 \mathcal{D}_s^2 + L\eta \|\xi\|_2^2 + L\eta\sqrt{\eta}\mathcal{D}_s \|\xi\|_2\right). \tag{2.58}$$

Standard tail bounds for χ^2-variables guarantee that

$$\mathbb{P}\left[\|\xi\|_2^2 \leq d\alpha_\varepsilon\right] \geq (1 - \varepsilon/16) \text{ for } \alpha_\varepsilon = 1 + 2\sqrt{\log(16/\varepsilon)} + 2\log(16/\varepsilon).$$

A simple observation reveals that the function $\tilde{w}$ defined in equation (2.29a) was chosen such that for any $\eta \leq \tilde{w}(s, \varepsilon)$, we have

$$L\eta^2\mathcal{D}_s^2 \leq \frac{\varepsilon}{128}, \quad L\eta d\alpha_\varepsilon \leq \frac{\varepsilon}{64}, \quad \text{and,} \quad L\eta\sqrt{\eta}\mathcal{D}_s\sqrt{d\alpha_\varepsilon} \leq \frac{\varepsilon}{128}.$$

Combining this observation with the high probability bound on $\|\xi\|_2$ and using the inequality (2.58) we obtain that $T \geq -\varepsilon/16$ with probability at least $1 - \varepsilon/16$. Plugging this bound in the inequality (2.56), we find that

$$\mathbb{P}\left[\frac{\Pi^*(z) \cdot \rho_z(x)}{\Pi^*(x) \cdot \rho_x(z)} \geq \exp\left(-\frac{\varepsilon}{16}\right)\right] \geq (1 - \varepsilon/16).$$

Thus, we have derived a desirable high probability lower bound on the accept-reject ratio. Substituting $\alpha = \exp(-\varepsilon/16)$ in the inequality (2.51) and using the fact that $e^{-\varepsilon/16} \geq 1 - \varepsilon/16$ for any scalar $\varepsilon > 0$, we find that

$$\mathbb{E}_{z\sim\mathcal{P}_x}\left[\min\left\{1, \frac{\Pi^*(z) \cdot \rho_z(x)}{\Pi^*(x) \cdot \rho_x(z)}\right\}\right] \geq 1 - \frac{\varepsilon}{8}, \quad \text{for any } \varepsilon \in (0,1) \text{ and } \eta \leq \tilde{w}(s,\varepsilon).$$

Substituting this bound in the inequality (2.50) completes the proof.

2.5.6 Proof of Theorem 2

The proof of this theorem is similar to the proof of Theorem 1. We begin by claiming that

$$d_{\mathrm{TV}}\left(\mathcal{P}_x^{\mathrm{MRW}(\eta)}, \mathcal{P}_y^{\mathrm{MRW}(\eta)}\right) = \frac{\varepsilon}{\sqrt{2}} \quad \text{for all } x, y \text{ such that } \|x-y\|_2 \leq \varepsilon\sqrt{\eta} \tag{2.59a}$$

$$d_{\mathrm{TV}}\left(\mathcal{P}_x^{\mathrm{MRW}(\eta)}, \mathcal{T}_x^{\mathrm{MRW}(\eta)}\right) = \frac{\varepsilon}{8} \quad \text{for all } x \in \mathcal{R}_s, \tag{2.59b}$$

for any $\eta \leq c\varepsilon^2 m/(\alpha_\varepsilon d^2 L^2 r(s))$ for some universal constant c. Plugging $s = \epsilon/(2\varpi)$, $\varepsilon = 1/2$ and arguing as in Section 2.5.2, we find that $\Phi_{\epsilon/2\varpi}^{\mathrm{MRW}(\eta)} \geq c'\sqrt{m\eta}$ for some universal constant c'. Using the convergence rate (2.26), we obtain that

$$d_{\mathrm{TV}}\left(\mathcal{T}_{\mathrm{MRW}(\eta)}^k(\mu_0), \Pi^*\right) \leq \varpi\frac{\epsilon}{2\varpi} + \varpi e^{-km\eta/c'} \leq \epsilon \quad \text{for all} \quad k \geq \frac{c'}{m\eta} \cdot \log\left(\frac{2\varpi}{\epsilon}\right), \tag{2.60}$$

for a suitably large constant c'. Substituting $\eta \leq cm/\left(d^2L^2r(\epsilon/2\varpi)\right)$, yields the claimed bound on mixing time of MRW.

It is now left to establish our earlier claims (2.59a) and (2.59b). Note that the initialization is normal distributed $\mathcal{P}_x^{\mathrm{MRW}(\eta)} = \mathcal{N}(x, 2\eta\mathbb{I}_d)$. For brevity, we drop the superscripts from our notations. Using the expression (2.47) for the KL-divergence and applying Pinsker's inequality leads to the upper bound

$$d_{\mathrm{TV}}\left(\mathcal{P}_x, \mathcal{P}_y\right) \leq \sqrt{2\mathrm{KL}(\mathcal{P}_x \parallel \mathcal{P}_y)} = \frac{\|x-y\|_2}{\sqrt{2\eta}},$$

which implies the claim (2.59a).

We now prove the bound (2.59b). Letting ρ_x to denote the density of the proposal distribution $\mathcal{P}_x$ and using the bounds (2.50) and (2.51), it suffices to prove that

$$\mathbb{P}_{z\sim\mathcal{P}_x}\left[\frac{\Pi^*(z)}{\Pi^*(x)} \geq \exp\left(-\frac{\varepsilon}{16}\right)\right] \overset{(i)}{=} \mathbb{P}_{z\sim\mathcal{P}_x}\left[f(x) - f(z) \geq -\frac{\varepsilon}{16}\right] \geq (1 - \varepsilon/16), \tag{2.61}$$

where step (i) follows from the fact that $\Pi^*(x) \propto e^{-f(x)}$. We have

$$\begin{aligned} f(x) - f(z) \overset{(i)}{\geq} \nabla f(z)^\top (x-z) &= \left(\nabla f(z) - \nabla f(x)\right)^\top (x-z) + \nabla f(x)^\top (x-z) \\ &\overset{(ii)}{\geq} -L\|x-z\|_2^2 + \nabla f(x)^\top (x-z) \\ &= -2L\eta\|\xi\|_2^2 + \sqrt{2\eta}\nabla f(x)^\top \xi \end{aligned} \tag{2.62}$$

where the step (i) follows from the convexity of the function f, step (ii) the smoothness of the function f. Note that the random variable $\chi := \nabla f(x)^\top \xi \sim \mathcal{N}(0, \|\nabla f(x)\|_2^2)$ and that $\|\nabla f(x)\|_2 \leq \mathcal{D}_s$ for any $x \in \mathcal{R}_s$. Consequently, we have $\chi \geq -\mathcal{D}_s \cdot 2\sqrt{\log(32/\varepsilon)}$

with probability at least $1 - \varepsilon/32$. On the other hand, using the standard tail bound for a Chi-squared random variable, we obtain that $\mathbb{P}\left[\|\xi\|_2^2 \geq d\alpha_\varepsilon\right] \leq \varepsilon/32$ for $\alpha_\varepsilon = 1+2\sqrt{\log(32/\varepsilon)}+2\log(32/\varepsilon)$. Recalling that $\mathcal{D}_s = L\sqrt{\frac{d}{m}}r(s)$ and doing straightforward calculation reveals that for $\eta \leq \frac{\varepsilon^2}{(8192\alpha_\varepsilon d\frac{L^2}{m}r(s))}$, we have

$$2L\eta d\alpha_\varepsilon \leq \frac{\varepsilon}{64} \quad \text{and} \quad \sqrt{2\eta}\mathcal{D}_s 2\sqrt{\log(32/\varepsilon)} \leq \frac{3\varepsilon}{64}$$

Combining these bounds with the high probability statements above and plugging in the inequality (2.62), we find that $f(x) - f(z) \geq -\varepsilon/16$ with probability at least $1 - \varepsilon/16$, which yields the claim (2.61).

2.6 Summary

In this chapter, we derived non-asymptotic bounds on the mixing time of the Metropolis adjusted Langevin algorithm and Metropolized random walk for log-concave distributions. These algorithms are based on a two-step scheme: (1) proposal step, and, (2) accept-reject step. Our results show that the accept-reject step, while it complicates the analysis, is practically very useful: algorithms involving this step mix significantly faster than the ones without it. In particular, we showed that for a strongly log-concave distribution in $\mathbb{R}^d$ with condition number κ, the ϵ-mixing time for MALA is of $O\left(d\kappa\log(1/\epsilon)\right)$. This guarantee significantly better than the $O\left(d\kappa^2/\epsilon^2\right)$ mixing time for ULA established in the literature. We also proposed a modified version of MALA to sample from non-strongly log-concave distributions and showed that it mixes in $O\left(d^3/\epsilon^{1.5}\right)$; thus, this algorithm dependency on the desired accuracy ϵ when compared to the $O\left(d^3/\epsilon^4\right)$ mixing time for ULA for the same task. Furthermore, we established $O\left(d\kappa^2\log(1/\epsilon)\right)$ mixing time bound for the Metropolized random walk for log-concave sampling.

Several fundamental questions arise from our work. All of our results are upper bounds on mixing time, and our simulation results suggest that they are tight for the choice of step size used in the Theorem statements. However, simulations from Section 2.4.4 suggest that potentially larger choices of step sizes are possible which would imply a faster mixing time, and consequently, providing theoretical guarantees for larger step sizes is a very interesting future direction. Furthermore, it remains to see if we can improve the mixing time of MALA from non-warm or deterministic start so that MALA is strictly better than ULA for any starting distribution. From a practitioner point of view, currently a hybrid algorithm seems to be a good middle ground: run ULA for a few steps to obtain moderate accuracy, and then employ ULA iterates to provide a warm start to MALA which would then generate highly accurate samples in reasonably few number of iterations.

Another open question is to rigorously determine the fundamental gap between the mixing times of first-order sampling methods and that of zeroth order sampling

methods. Noting that MALA is a first-order method while MRW is a zeroth order method, from our work, we obtain that two class of methods differ in a factor of the condition number κ of the target distribution. It is an exciting question to determine if this gap is tight between these two class of sampling methods.

Chapter 3

Hamiltonian Monte Carlo

In this chapter, we consider the state-of-the-art Markov chain Monte Carlo sampling algorithm, Hamiltonian Monte Carlo, for drawing samples over unconstrained state space. More specifically, we focus on its most widely used variant, the Metropolized HMC with the Störmer-Verlet or leapfrog integrator. First, we provide a non-asymptotic upper bound on the mixing time of the Metropolized HMC with explicit choices of step-size and number of leapfrog steps. This bound gives a precise quantification of the faster convergence of Metropolized HMC relative to simpler MCMC algorithms such as the Metropolized random walk, or Metropolized Langevin algorithm. Second, we provide a general framework for sharpening mixing time bounds Markov chains initialized at a substantial distance from the target distribution over continuous spaces. We apply this sharpening device to the Metropolized random walk and Langevin algorithms, thereby obtaining improved mixing time bounds from a non-warm initial distribution. This strictly improves upon the conductance-based proof techniques introduced in the previous chapter.

3.1 Introduction

As we have seen in the previous chapter, there are a variety of MCMC methods for sampling from target distributions with smooth densities [160, 164, 165, 23]. Among them, the method of Hamiltonian Monte Carlo (HMC) stands out among practitioners: it is the default sampler in many popular software packages, including Stan [31], Mamba [181], and Tensorflow [130]. We refer the reader to the papers [145, 79, 57] for further examples and discussion of the HMC method. There are a number of variants of HMC, but the most popular choice involves combination of the leapfrog integrator with Metropolis-Hastings correction. Throughout this chapter, we reserve the terminology HMC to refer to this particular Metropolized algorithm. The idea of using Hamiltonian dynamics in simulation first appeared in Alder and Wainwright [3]. Duane et al. [54] introduced MCMC with Hamiltonian dynamics, and referred to it as Hybrid Monte Carlo. The algorithm was further refined by Neal [144], and later re-christened

in statistics community as Hamiltonian Monte Carlo. We refer the reader to Neal [145] for an illuminating overview of the history of HMC and discussion of contemporary work.

While HMC enjoys fast convergence in practice, a theoretical understanding of this behavior remains incomplete. Some intuitive explanations are based on its ability to maintain a constant asymptotic accept-reject rate with large step-size (e.g. [42]). Others (e.g. Neal [145]) suggest, based on intuition from the continuous-time limit of the Hamiltonian dynamics, that HMC is able to suppress random walk behavior using momentum. However, these intuitive arguments do not provide rigorous or quantitative justification for the fast convergence of the discrete-time HMC used in practice.

More recently, general asymptotic conditions under which HMC will or will not be geometrically ergodic have been established in some recent papers [57, 114]. Other work has yielded some insight into the mixing properties of different variants of HMC, but it has focused mainly on *unadjusted* versions of the algorithm. Mangoubi et al. [127, 128] study versions of unadjusted HMC based on Euler discretization or leapfrog integrator (but omitting the Metropolis-Hastings step), and provide explicit bounds on the mixing time as a function of dimension d, condition number κ and error tolerance $\epsilon > 0$. Lee and Vempala [108] studied an extended version of HMC that involves applying an ordinary differential equation (ODE) solver; they established bounds with sublinear dimension dependence, and even polylogarithmic for certain densities (e.g., those arising in Bayesian logistic regression). The mixing time for the same algorithm is further refined in the recent work by Chen and Vempala [37]. In a similar spirit, Lee and Vempala [112] studied the Riemannian variant of HMC (RHMC) with an ODE solver focusing on sampling uniformly from a polytope. While their result could be extended to log-concave sampling, the practical implementation for log-concave sampling of their ODE solver is unclear, and moreover requires a regularity condition on all the derivatives of density. It should be noted that such unadjusted HMC methods behave differently from the Metropolized version most commonly used in practice. In the absence of the Metropolis-Hastings correction, the resulting Markov chain no longer converges to the correct target distribution, but instead exhibits a persistent bias even in the limit of infinite iterations. Consequently, analysis of such sampling methods requires controlling this bias; doing so leads to mixing times that scale polynomially in $1/\epsilon$, in sharp contrast with the $\log(1/\epsilon)$ that is typical for Metropolis-Hastings corrected methods.

Most closely related to our work is the recent work by Bou-Rabee et al. [17], which studies the same Metropolized HMC algorithm that we analyze in this chapter. These authors use coupling methods to analyze HMC for a class of distributions that are strongly log-concave outside of a compact set. In the strongly log-concave case, they prove a mixing time bound that scales at least as $d^{3/2}$ in the dimension d. It should be noted that with a "warm" initialization, this dimension dependence grows more quickly than known bounds for the MALA algorithm [58, 60], and so does not explain the superiority of HMC in practice.

In practice, it is known that Metropolized HMC is fairly sensitive to the choice of its parameters, namely the step-size η used in the discretization scheme, and the

number of leapfrog steps K. At one extreme, taking a single leapfrog step $K = 1$, the algorithm reduces to the Metropolis adjusted Langevin algorithm (MALA). More generally, if too few leapfrog steps are taken, then HMC is likely to exhibit a random walk behavior similar to MALA. At the other extreme, if K is too large, the leapfrog steps tend to wander back to a neighborhood of the initial state, which leads to wasted computation as well as slower mixing [14]. In terms of the step size η, choosing an overly large step size makes the discretization diverge from the underlying continuous dynamics, and causes the Metropolis acceptance probability to drop, hence slowing down the algorithm. On the other hand, an overly small choice of η does not allow the algorithm to explore the state space rapidly enough. Various automatic strategies for tuning these two parameters, involving heuristics and additional computational cost, have been proposed [195, 79, 200]. Among these strategies, the No-U-Turn (NUTS) sampler [79], is one of the most popular, used by default in the Stan package [31].

Past work on mixing time dependency on initialization: Many proof techniques for the convergence of continuous-state Markov chains are inspired by the large body of work on discrete-state Markov chains; for instance, see the surveys [116, 4] and references therein. Historically, much work has been devoted to improving the mixing time dependency on the initial distribution. For discrete-state Markov chains, Diaconis and Saloff-Coste [52] were the first to show that the logarithmic dependency of the mixing time of a Markov chain on the warmness parameter[1] of the starting distribution can be improved to double-logarithmic. This improvement—from logarithmic to doubly logarithmic—allows for a good bound on the mixing time even when starting distribution is not available. The innovation underlying this improvement is the use of log-Sobolev inequalities in place of the usual isoperimetric inequality. Later, closely related ideas such as average conductance [117, 94], evolving sets [137] and spectral profile [71] were shown to be effective for reducing dependence on initial conditions for discrete space chains. Thus far, only the notion of average conductance [117, 94] has been adapted to continuous-state Markov chains so as to sharpen mixing time analysis of the Ball walk [119].

Our contributions: This chapter makes two primary contributions. First, we provide a non-asymptotic upper bound on the mixing time of the Metropolized HMC algorithm for smooth densities (see Theorem 3). This theorem applies to the form of Metropolized HMC (based on the leapfrog integrator) that is most widely used in practice. To the best of our knowledge, Theorem 3 is the first rigorous confirmation of the faster non-asymptotic convergence of the Metropolized HMC as compared to MALA and other simpler Metropolized algorithms.[2] Other related works on HMC consider either its unadjusted version (without accept-reject step) with different integrators [127,

[1] See equation (2.6) for a formal definition.

[2] As noted earlier, previous results by Bou-Rabee et al. [17] on Metropolized HMC do not establish that it mixes more rapidly than MALA.

128] or the HMC based on an ODE solver [108, 112]. While the dimension dependency for these algorithms is usually better than MALA, they have polynomial dependence on the target error ϵ while MALA's mixing time scales as $\log(1/\epsilon)$. Moreover, our direct analysis of the Metropolized HMC with a leapfrog integrator provides explicit choices of the hyper-parameters for the sampler, namely, the step-size and the number of leapfrog updates in each step. Our theoretical choices of the hyper-parameters could potentially reduce the difficulty of parameter tuning in practical HMC implementations.

Our second main contribution is formalized in Lemmas 4 and 5: we develop results based on the conductance profile in order to prove quantitative convergence guarantees general continuous state space Markov chains. Doing so involves non-trivial extensions of ideas from discrete state Markov chains to those in continuous state spaces. Our results not only enable us to establish the mixing time bounds for HMC with different classes of target distributions, but also allow simultaneous improvements on mixing time bounds of several Markov chains (for general continuous-state space) when the starting distribution is far from the stationary distribution. Consequentially, we improve upon previous mixing time bounds for Metropolized Random Walk (MRW) and MALA [58], when the starting distribution is not *warm* with respect to the target distribution (see Theorem 4).

While this high-level road map is clear, a number of technical challenges arise en route in particular in controlling the conductance profile of HMC. The use of multiple gradients in HMC helps it mix faster but also complicates the analysis; in particular, a key step is to control the overlap between the transition distributions of HMC chain at two nearby points; doing so requires a delicate argument (see Lemma 6 and Section 3.5.3 for further details).

Table 3.1 provides an informal summary of our mixing time bounds of HMC and how they compare with known bounds for MALA when applied to log-concave target distributions. From the table, we see that Metropolized HMC takes fewer gradient evaluations than MALA to mix to the same accuracy for log-concave distributions. Note that our current analysis establishes logarithmic dependence on the target error ϵ for strongly-log-concave as well as for a sub-class of weakly log-concave distributions. [3]

Organization: The remainder of the chapter is organized as follows. Section 3.2 is devoted to background on the idea of Monte Carlo approximation, Markov chains and MCMC algorithms, and the introduction of the HMC algorithm. Section 3.3 contains our main results on mixing time of HMC in Section 3.3.2, followed by the general framework for obtaining sharper mixing time bounds in Section 3.3.3 and its application to MALA and MRW in Section 3.3.4. In Section 3.4, we describe some numerical experiments that we performed to explore the sharpness of our theoretical predictions in some simple scenarios. In Section 3.5, we prove Theorem 3 and Corollary 6, with the

[3]For a comparison with previous results on unadjusted HMC or ODE based HMC refer to the discussion after Corollary 3 and Table A.4 in Appendix A.4.2.

Sampling algorithm	Strongly log-concave Assumption (B) ($\kappa \ll d$)	Weakly log-concave Assumption (C)	Assumption (D)
MALA (improved bound in Thm 4 in this chapter)	$d\kappa \log \frac{1}{\epsilon}$ [58]	$\frac{d^2}{\epsilon^{\frac{3}{2}}} \log \frac{1}{\epsilon}$ [58]	$d^{\frac{3}{2}} \log \frac{1}{\epsilon}$ [129]
Metropolized HMC with leapfrog integrator [this chapter]	$d^{\frac{11}{12}} \kappa \log \frac{1}{\epsilon}$ (Corollary 3)	$\frac{d^{\frac{11}{6}}}{\epsilon} \log \frac{1}{\epsilon}$ (Corollary 7)	$d^{\frac{4}{3}} \log \frac{1}{\epsilon}$ (Corollary 7)

Table 3.1. Comparisons of the number of gradient evaluations needed by MALA and Metropolized HMC with leapfrog integrator from a *warm start* to obtain an ϵ-accurate sample in TV distance from a log-concave target distribution on $\mathbb{R}^d$. The second column corresponds to strongly log-concave densities with condition number κ, and the third and fourth column correspond to weakly log-concave densities satisfying certain regularity conditions.

proofs of technical lemmas and other results deferred to the appendices. We conclude in Section 3.6 with a discussion of our results and future directions.

3.2 Background

In this section, we describe several MCMC algorithms, including the Metropolized random walk (MRW), the Metropolis-adjusted Langevin algorithm (MALA), and the Metropolis-adjusted Hamiltonian Monte Carlo (HMC) algorithm. Readers familiar with the literature may skip directly to the Section 3.3, where we set up and state our main results.

3.2.1 MRW and MALA algorithms

One of the simplest Markov chain algorithms for sampling from a density of the form (2.1) defined on $\mathbb{R}^d$ is the Metropolized random walk (MRW). Given state $x_i \in \mathbb{R}^d$ at iterate i, it generates a new proposal vector $z_{i+1} \sim \mathcal{N}(x_i, 2\eta \mathbb{I}_d)$, where $\eta > 0$ is a step-size parameter.[4] It then decides to accept or reject z_{i+1} using a Metropolis-Hastings correction; see Algorithm 1 for the details. Note that the MRW algorithm uses information about the function f only via querying function values, but not the gradients.

The Metropolis-adjusted Langevin algorithm (MALA) is a natural extension of the MRW algorithm: in addition to the function value $f(\cdot)$, it also assumes access to its gradient $\nabla f(\cdot)$ at any state $x \in \mathbb{R}^d$. Given state x_i at iterate i, it observes $(f(x_i), \nabla f(x_i))$

[4]The factor 2 in the stepsize definition is a convenient notational choice so as to facilitate comparisons with other algorithms.

and then generates a new proposal $z_{i+1} \sim \mathcal{N}(x_i - \eta \nabla f(x_i), 2\eta \mathbb{I}_d)$, followed by a suitable Metropolis-Hastings correction; see Algorithm 2 for the details. The MALA algorithm has an interesting connection to the Langevin diffusion, a stochastic process whose evolution is characterized by the stochastic differential equation (SDE)

$$dX_t = -\nabla f(X_t) + \sqrt{2} dW_t. \tag{3.1}$$

The MALA proposal can be understood as the Euler-Maruyama discretization of the SDE (3.1).

3.2.2 HMC sampling

The HMC sampling algorithm from the physics literature was introduced to the statistics literature by Neal; see his survey [145] for the historical background. The method is inspired by Hamiltonian dynamics, which describe the evolution of a state vector $q(t) \in \mathbb{R}^d$ and its momentum $p(t) \in \mathbb{R}^d$ over time t based on a Hamiltonian function $\mathcal{H} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ via Hamilton's equations:

$$\frac{dq}{dt}(t) = \frac{\partial \mathcal{H}}{\partial p}(p(t), q(t)), \quad \text{and} \quad \frac{dp}{dt}(t) = -\frac{\partial \mathcal{H}}{\partial q}(p(t), q(t)). \tag{3.2}$$

A straightforward calculation using chain rule shows that the Hamiltonian remains invariant under these dynamics—that is, $\mathcal{H}(p(t), q(t)) = C$ for all $t \in \mathbb{R}$. A typical choice of the Hamiltonian $\mathcal{H} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is given by

$$\mathcal{H}(p, q) = f(q) + \frac{1}{2} \|p\|_2^2 . \tag{3.3}$$

The ideal HMC algorithm for sampling is based on the continuous Hamiltonian dynamics; as such, it is not implementable in practice, but instead a useful algorithm for understanding. For a given time $T > 0$ and vectors $u, v \in \mathbb{R}^d$, let $q_T(u, v)$ denote the q-solution to Hamilton's equations at time T and with initial conditions $(p(0), q(0)) = (u, v)$. At iteration k, given the current iterate X_k, the ideal HMC algorithm generates the next iterate X_{k+1} via the update rule $X_{k+1} = q_T(p_k, X_k)$ where $p_k \sim N(0, \mathbb{I}_d)$ is a standard normal random vector, independent of X_k and all past iterates. It can be shown that with an appropriately chosen T, the ideal HMC algorithm converges to the stationary distribution π^* without a Metropolis-Hastings adjustment (see [145, 128] for the existence of such solution and its convergence).

However, in practice, it is impossible to compute an exact solution to Hamilton's equations. Rather, one must approximate the solution $q_T(p_k, X_k)$ via some discrete process. There are many ways to discretize Hamilton's equations other than the simple Euler discretization; see Neal [145] for a discussion. In particular, using the leapfrog or Störmer-Verlet method for integrating Hamilton's equations leads to the Hamiltonian Monte Carlo (HMC) algorithm. It simulates the Hamiltonian dynamics for K steps

via the leapfrog integrator. At each iteration, given previous state q_0 and fresh $p_0 \sim \mathcal{N}(0, \mathbb{I}_d)$, it runs the following updates for K times, for $0 \leq k \leq K-1$,

$$p_{k+\frac{1}{2}} = p_k - \frac{\eta}{2}\nabla f(q_k) \tag{3.4a}$$

$$q_{k+1} = q_k + \eta p_{k+\frac{1}{2}} \tag{3.4b}$$

$$p_{k+1} = p_{k+\frac{1}{2}} - \frac{\eta}{2}\nabla f(q_{k+1}). \tag{3.4c}$$

Since discretizing the dynamics generates discretization error at each iteration, it is followed by a Metropolis-Hastings adjustment where the proposal (p_K, q_K) is accepted with probability

$$\min\left\{1, \frac{\exp\left(-\mathcal{H}(p_K, q_K)\right)}{\exp\left(-\mathcal{H}(p_0, q_0)\right)}\right\}. \tag{3.5}$$

See Algorithm 3 for a detailed description of the HMC algorithm with leapfrog integrator.

Algorithm 3: Metropolized HMC with leapfrog integrator

Input: Step size η, number of internal leapfrog updates K, and a sample x_0 from a starting distribution μ_0

Output: Sequence $x_1, x_2, \ldots$

for $i = 0, 1, \ldots$ **do**
  Proposal step:
  $q_0 \leftarrow x_i$
  Draw $p_0 \sim \mathcal{N}(0, \mathbb{I}_d)$
  for $k = 1, \ldots, K$ **do**
    $(p_k, q_k) \leftarrow \text{Leapfrog}(p_{k-1}, q_{k-1}, \eta)$
  end
  % q_K is now the new proposed state
  Accept-reject step:
    compute $\alpha_{i+1} \leftarrow \min\left\{1, \frac{\exp\left(-\mathcal{H}(p_K, q_K)\right)}{\exp\left(-\mathcal{H}(p_0, q_0)\right)}\right\}$
    With probability α_{i+1} *accept* the proposal: $x_{i+1} \leftarrow q_K$
    With probability $1 - \alpha_{i+1}$ *reject* the proposal: $x_{i+1} \leftarrow x_i$
end
Program `Leapfrog`(p, q, η)**:**
  $\tilde{p} \leftarrow p - \frac{\eta}{2}\nabla f(q)$
  $\tilde{q} \leftarrow q + \eta\tilde{p}$
  $\tilde{p} \leftarrow \tilde{p} - \frac{\eta}{2}\nabla f(\tilde{q})$
return $(\tilde{p}, \tilde{q})$

Remark: The HMC with leapfrog integrator can also be seen as a multi-step version of a simpler Langevin algorithm. Indeed, running the HMC algorithm with $K = 1$ is equivalent to the MALA algorithm after a re-parametrization of the step-size η. In practice, one also uses the HMC algorithm with a modified Hamiltonian, in which the quadratic term $\|p\|_2^2$ is replaced by a more general quadratic form $p^T M p$. Here M is a symmetric positive definite matrix to be chosen by the user; see Appendix A.4.1 for further discussion of this choice. In the main text, we restrict our analysis to the case $M = I$.

3.3 Convergence of Hamiltonian Monte Carlo

We now turn to the statement of our main results. We remind the readers that HMC refers to Metropolized HMC with leapfrog integrator, unless otherwise specified. We begin in Section 3.3.2 with our results for HMC: first, we derive the mixing time bounds for general target distributions in Theorem 3 and then apply that result to obtain concrete guarantees for HMC with strongly log-concave target distributions. We defer the discussion of weakly log-concave target distributions and perturbations of log-concave distributions to Appendix A.3.

In Section 3.3.3, we discuss the underlying results that are used to derive sharper mixing time bounds using conductance profile (see (Lemmas 4 and 5). In addition to being central to the proof of Theorem 3 in Section 3.5, these lemmas also allow us to sharpen mixing time guarantees for MALA and MRW (without much work). We state these improvements in Section 3.3.4.

3.3.1 Assumptions on the target distribution

In this section, we introduce some regularity notions and state the assumptions on the target distribution that our results in the next section rely on.

Regularity conditions: Recall from Section 1.5 that a function f is called:

$$L\text{-smooth}: \quad f(y) - f(x) - \nabla f(x)^\top (y-x) \le \frac{L}{2}\|x-y\|_2^2 \tag{3.6a}$$

$$m\text{-strongly convex}: \quad f(y) - f(x) - \nabla f(x)^\top (y-x) \ge \frac{m}{2}\|x-y\|_2^2 \tag{3.6b}$$

$$L_{\mathrm{H}}\text{-Hessian Lipschitz}: \quad \left\|\nabla^2 f(x) - \nabla^2 f(y)\right\|_2 \le L_{\mathrm{H}}\|x-y\|_2, \tag{3.6c}$$

where in all cases, the inequalities hold for all $x, y \in \mathbb{R}^d$.

A distribution Π with support $\mathcal{X} \subset \mathbb{R}^d$ is said to satisfy the *isoperimetric inequality* ($\mathfrak{a} = 0$) or the *log-isoperimetric inequality* ($\mathfrak{a} = \frac{1}{2}$) with constant $\psi_{\mathfrak{a}}$ if given any partition S_1, S_2, S_3 of $\mathcal{X}$, we have

$$\Pi(S_3) \ge \frac{1}{2\psi_{\mathfrak{a}}} \cdot d(S_1, S_2) \cdot \min\{\Pi(S_1), \Pi(S_2)\} \cdot \log^{\mathfrak{a}}\left(1 + \frac{1}{\min\{\Pi(S_1), \Pi(S_2)\}}\right). \tag{3.6d}$$

where the distance between two sets S_1, S_2 is defined as $d(S_1, S_2) = \inf_{x \in S_1, y \in S_2} \{\|x - y\|_2\}$. For a distribution Π with density π and a given set Ω, its restriction to Ω is the distribution Π_Ω with the density $\pi_\Omega(x) = \frac{\pi(x)\mathbf{1}_\Omega(x)}{\Pi(\Omega)}$.

Assumptions on the target distribution: We introduce two sets of assumptions for the target distribution:

(A) We say that the target distribution Π^* is $(L, L_{\mathrm{H}}, s, \psi_{\mathfrak{a}}, M)$-*regular* if the negative log density f is L-smooth (3.6a) and has L_{H}-Lipschitz Hessian (3.6c), and there exists a convex measurable set Ω such that the distribution Π^*_Ω is $\psi_{\mathfrak{a}}$-isoperimetric (3.6d), and the following conditions hold:

$$\Pi^*(\Omega) \geq 1 - s \quad \text{and} \quad \|\nabla f(x)\|_2 \leq M, \quad \text{for all } x \in \Omega. \tag{3.6e}$$

(B) We say that the target distribution Π^* is (L, L_{H}, m)-*strongly log-concave* if the negative log density is L-smooth (3.6a), m-strongly convex (3.6b), and L_{H}-Hessian-Lipschitz (3.6c). Moreover, we use $x^\star$ to denote the unique mode of Π^* whenever f is strongly convex.

Assumption (B) has appeared in several past papers on Langevin algorithms [43, 58, 38] and the Lipschitz-Hessian condition (3.6c) has been used in analyzing Langevin algorithms with inaccurate gradients [44] as well as the unadjusted HMC algorithm [128]. It is worth noting Assumption (A) is strictly weaker than Assumption (B), since it allows for distributions that are not log-concave. As we show in Lemma 20, Assumption (B) implies a version of Assumption (A); see Appendix A.2 for details.

3.3.2 Mixing time bounds for HMC

We start with the mixing time bound for HMC applied to any distribution Π^* satisfying Assumption (A). Let HMC-(K, η) denote the $\frac{1}{2}$-lazy Metropolized HMC algorithm with η step size and K leapfrog steps in each iteration. Let $\tau_2^{\mathrm{HMC}}(\epsilon; \mu_0)$ denote the $\mathcal{L}_2$-mixing time (2.5a) for this chain with the starting distribution μ_0.

Theorem 3. *Consider an $(L, L_H, s, \psi_{\mathfrak{a}}, M)$-regular target distribution (cf. Assumption (A)) and a ϖ-warm initial distribution μ_0. Then for any fixed target accuracy $\epsilon \in (0, 1)$ such that $\epsilon^2 \geq 2\varpi s$, there exist choices of the parameters (K, η) such that HMC-(K, η)chain with μ_0 start satisfies*

$$\tau_2^{HMC}(\epsilon; \mu_0) \leq \begin{cases} c \cdot \max\left\{\log \varpi, \frac{\psi_{\mathfrak{a}}^2}{K^2\eta^2} \log\left(\frac{\log \varpi}{\epsilon}\right)\right\} & \text{if } \mathfrak{a} = \frac{1}{2} \qquad \text{[log-isoperi (3.6d)]} \\ c \cdot \frac{\psi_{\mathfrak{a}}^2}{K^2\eta^2} \log\left(\frac{\varpi}{\epsilon}\right) & \text{if } \mathfrak{a} = 0 \qquad \text{[isoperi (3.6d)].} \end{cases}$$

See Section 3.5.2 for the proof, where we also provide explicit conditions on η and K in terms of the other parameters (cf. equation (3.22b)).

Theorem 3 covers mixing time bounds for distributions that satisfy isoperimetric or log-isoperimetric inequality provided that: (a) both the gradient and Hessian of the negative log-density are Lipschitz; and (b) there is a convex set that contains a large mass $(1 - s)$ of the distribution. The mixing time only depends on two quantities: the log-isoperimetric (or isoperimetric) constant of the target distribution and the effective step-size $K^2\eta^2$. As shown in the sequel, these conditions hold for log-concave distributions as well as certain perturbations of them. If the distribution satisfies a log-isoperimetric inequality, then the mixing time dependency on the initialization warmness parameter ϖ is relatively weak $O(\log\log\varpi)$. On the other hand, when only an isoperimetric inequality (but not log-isoperimetric) is available, the dependency is relatively larger $O(\log\varpi)$. In our current analysis, we can establish the ϵ-mixing time bounds up-to an error ϵ such that $\epsilon^2 \geq 2\varpi s$. If mixing time bounds up to an arbitrary accuracy are desired, then the distribution needs to satisfy (3.6e) for arbitrary small s. For example, as we later show in Lemma 20, arbitrary small s can be imposed for strongly log-concave densities (i.e. satisfying Assumption (B)).

Let us now derive several corollaries of Theorem 3. We begin with non-asymptotic mixing time bounds for HMC-(K, η) chain for strongly-log concave target distributions. Then we briefly discuss the corollaries for weakly log-concave target and non-log-concave target distributions and defer the precise statements to Appendix A.3. These results also provide a basis for comparison of our results with prior work.

Strongly log-concave target

We now state an explicit mixing time bound of HMC for a strongly log-concave distribution. We consider an (L, L_{H}, m)-strongly log-concave distribution (assumption (B)). We use $\kappa = L/m$ to denote the condition number of the distribution. Our result makes use of the following function

$$r(s) := 1 + \max\left\{\left(\frac{\log(1/s)}{d}\right)^{1/4}, \left(\frac{\log(1/s)}{d}\right)^{1/2}\right\}, \tag{3.7a}$$

and involves the stepsize choices

$$\eta_{\mathrm{warm}} = \sqrt{\frac{1}{cL \cdot r\left(\frac{\epsilon^2}{2\varpi}\right) d^{\frac{7}{6}}}}, \quad \text{and} \quad \eta_{\mathrm{feas}} = \sqrt{\frac{1}{cL \cdot r\left(\frac{\epsilon^2}{2\kappa^d}\right)} \min\left\{\frac{1}{d\kappa^{\frac{1}{2}}}, \frac{1}{d^{\frac{2}{3}}\kappa^{\frac{5}{6}}}, \frac{1}{d^{\frac{1}{2}}\kappa^{\frac{3}{2}}}\right\}}, \tag{3.7b}$$

With these definitions, we have the following:

Corollary 3. *Consider an (L, L_H, m)-strongly log-concave target distribution Π^* (cf. Assumption (B)) such that $L_H^{2/3} = O(L)$, and any error tolerance $\epsilon \in (0, 1)$.*

(a) *Suppose that* $\kappa = O(d^{\frac{2}{3}})$ *and* $\varpi = O\left(\exp\left(d^{\frac{2}{3}}\right)\right)$. *Then with any* ϖ*-warm initial distribution* μ_0, *hyper-parameters* $K = d^{\frac{1}{4}}$ *and* $\eta = \eta_{warm}$, *the HMC-*(K, η) *chain satisfies*

$$\tau_2^{HMC}(\epsilon; \mu_0) \leq c\, d^{\frac{2}{3}}\, \kappa\, r\left(\frac{\epsilon^2}{2\varpi}\right) \log\left(\frac{\log \varpi}{\epsilon}\right). \tag{3.8a}$$

(b) *With the initial distribution* $\mu_\dagger = \mathcal{N}(x^\star, \frac{1}{L}\mathbb{I}_d)$, *hyper-parameters* $K = \kappa^{\frac{3}{4}}$ *and* $\eta = \eta_{feas}$, *the HMC-*(K, η) *chain satisfies*

$$\tau_2^{HMC}(\epsilon; \mu_\dagger) \leq c\, r\left(\frac{\epsilon^2}{2\kappa^d}\right) \max\left\{d \log \kappa, \max\left[d, d^{\frac{2}{3}}\kappa^{\frac{1}{3}}, d^{\frac{1}{2}}\kappa\right] \log\left(\frac{d \log \kappa}{\epsilon}\right)\right\}. \tag{3.8b}$$

See Appendix A.2 for the proof. In the same appendix, we also provide a more refined mixing time of the HMC chain for a more general choice of hyper-parameters (see Corollary 6). In fact, as shown in the proof, the assumption $L_{\mathrm{H}}^{2/3} = O(L)$ is not necessary in order to control mixing; rather, we adopted it above to simplify the statement of our bounds. A more detailed discussion on the particular choice for step size η is provided in Appendix A.4.

MALA vs HMC—Warm start: Corollary 3 provides mixing time bounds for two cases. The first result (3.8a) implies that given a warm start (with constant ϖ) for a well-conditioned strongly log concave distribution ($\kappa \ll d$), the ϵ-$\mathcal{L}_2$-mixing time[5] of HMC scales $\widetilde{O}(d^{\frac{2}{3}} \log(1/\epsilon))$. It is interesting to compare this guarantee with known bounds for the MALA algorithm; however, in order to do so in a fair way, we need to track the total number of gradient evaluation required by the HMC-(K, η) chain to mix. (Recall that each iteration of MALA uses only a single gradient evaluation.) For HMC to achieve accuracy ϵ, the total number of gradient evaluations is given by $K \cdot \tau_2^{\mathrm{HMC}}(\epsilon; \mu_0)$, which (in this case), scales as $\widetilde{O}(d^{\frac{11}{12}}\kappa \log(1/\epsilon))$. (This rate was also summarized in Table 3.1.) Note that the corresponding number of gradient evaluations for MALA (Theorem 1 [58]) is $\widetilde{O}(d\kappa \log(1/\epsilon))$. As a result, we conclude that for this case, the upper bound for HMC is $d^{\frac{1}{12}}$ better than the known upper bound for MALA. We summarize the rates for this case in Table 3.2. Note that MRW is a zeroth order algorithm and does not make use of gradient information.

MALA vs HMC—Feasible start: In the second result (3.8b), we cover the case when a warm start is not available. In particular, we analyze the HMC chain with

[5]Note that $r(\epsilon^2) \leq 6$ for $\epsilon \geq \frac{2}{e^{d/2}}$ and thus we can treat r as a small constant for a large range of ϵ. Otherwise, if ϵ needs to be extremely small, the results still hold with an extra $\log^{\frac{1}{2}}\left(\frac{1}{\epsilon}\right)$ dependency.

Sampling algorithm	Mixing time	#Gradient evaluations
MRW [58, Theorem 2]	$d\kappa^2 \cdot \log \frac{1}{\epsilon}$	NA
MALA [58, Theorem 1]	$d\kappa \cdot \log \frac{1}{\epsilon}$	$d\kappa \cdot \log \frac{1}{\epsilon}$
HMC-(K, η) [ours, Corollary 3]	$d^{\frac{2}{3}}\kappa \cdot \log \frac{1}{\epsilon}$	$d^{\frac{11}{12}}\kappa \cdot \log \frac{1}{\epsilon}$

Table 3.2. Summary of the ϵ-mixing time and the corresponding number of gradient evaluations for MRW, MALA and HMC from a *warm start* with an (L, L_{H}, m)-strongly-log-concave target. These statements hold under the assumption $L_{\mathrm{H}}^{2/3} = O(L)$, $\kappa = \frac{L}{m} \ll d$, and omit logarithmic terms in dimension.

the feasible initial distribution $\mu_\dagger = \mathcal{N}(x^\star, \frac{1}{L}\mathbb{I}_d)$. Here $x^\star$ denotes the unique mode of the target distribution and can be easily computed using an optimization scheme like gradient descent. It is not hard to show (see Corollary 1 in Dwivedi et al. [58]) that for an L-smooth (3.6a) and m strongly log-concave target distribution (3.6b), the distribution $\mu_\dagger$ acts as a $\kappa^{d/2}$-warm start distribution. Once again, it is of interest to determine whether HMC takes fewer gradient steps when compared to MALA to obtain an ϵ-accurate sample. We summarize the results in Table 3.3 (where log factors are hidden) and note that HMC with $K = \kappa^{3/4}$ is faster than MALA for as long as κ is not too large. From the last column, we find that when $\kappa \ll d^{\frac{1}{2}}$, HMC is faster than MALA by a factor of $\kappa^{\frac{1}{4}}$ in terms of number of gradient evaluations.[6]

Sampling algorithm	Mixing time	# Gradient Evaluations	
		general κ	$\kappa \ll d^{\frac{1}{2}}$
MRW [ours, Theorem 4]	$d\kappa^2$	NA	NA
MALA [ours, Theorem 4]	$\max\left\{d\kappa, d^{\frac{1}{2}}\kappa^{\frac{3}{2}}\right\}$	$\max\left\{d\kappa, d^{\frac{1}{2}}\kappa^{\frac{3}{2}}\right\}$	$d\kappa$
HMC-(K, η) [ours, Corollary 3]	$\max\left\{d, d^{\frac{2}{3}}\kappa^{\frac{1}{3}}, d^{\frac{1}{2}}\kappa\right\}$	$\max\left\{d\kappa^{\frac{3}{4}}, d^{\frac{2}{3}}\kappa^{\frac{13}{12}}, d^{\frac{1}{2}}\kappa^{\frac{7}{4}}\right\}$	$d\kappa^{\frac{3}{4}}$

Table 3.3. Summary of the ϵ-mixing time and the corresponding number of gradient evaluations for MRW, MALA and HMC from the *feasible start* $\mu_\dagger = \mathcal{N}(x^\star, \frac{1}{L}\mathbb{I}_d)$ for an (L, L_{H}, m)-strongly-log-concave target. Here $x^\star$ denotes the unique mode of the target distribution. These statements hold uner the assumption $L_{\mathrm{H}} = O(L^{\frac{3}{2}})$, and hide the logarithmic factors in ϵ, d and $\kappa = L/m$.

[6] It is worth noting that for the feasible start $\mu_\dagger$, the mixing time bounds for MALA and MRW in our prior work [58] were loose by a factor d when compared to the tighter bounds in Theorem 4 derived later in this chapter.

Metropolized HMC vs Unadjusted HMC: There are many recent results on the 1-Wasserstein distance mixing of unadjusted versions of HMC (for instance, see the papers e.g. [128, 108]). For completeness, we compare our results with them in the Appendix A.4.2; in particular, see Table A.4 for a comparative summary.) We remark that comparisons of these different results is tricky for two reasons: (a) The 1-Wasserstein distance and the total variation distance are not strictly comparable, and, (b) the unadjusted HMC results always have a polynomial dependence on the error parameter ϵ while our results for Metropolized HMC have a superior logarithmic dependence on ϵ. Nonetheless, the second difference between these chains has a deeper consequence, upon which we elaborate further in Appendix A.4.2. On one hand, the unadjusted chains have better mixing time in terms of scaling with d, if we fix ϵ or view it as independent of d. On the other hand, when such chains are used to estimate certain higher-order moments, the polynomial dependence on ϵ might become the bottleneck and Metropolis-adjusted chains would become the method of choice.

Ill-conditioned target distributions: In order to keep the statement of Corollary 3 simple, we stated the mixing time bounds of HMC-(K, η)-chain only for a particular choice of (K, η). In our analysis, this choice ensures that HMC is better than MALA only when condition number κ is small. For Ill-conditioned distributions, i.e., when κ is large, finer tuning of HMC-(K, η)-chain is required. In Appendices A.2 and A.4 (see Table A.1 for the hyper-parameter choices), we show that HMC is strictly better than MALA as long as $\kappa \leq d$ and as good as MALA when $\kappa \geq d$.

Beyond strongly-log-concave: The proof of Corollary 3 is based on the fact that (L, L_{H}, m)-strongly-log-concave distribution is in fact an $(L, L_{\mathrm{H}}, s, \psi_{1/2}, M_s)$ -regular distribution for any $s \in (0, 1)$. Here $\psi_{1/2} = 1/\sqrt{m}$ is fixed and the bound on the gradient $M_s = r(s)\sqrt{d/m}$ depends on the choice of s. The result is formally stated in Lemma 20 in Appendix A.2. Moreover, in Appendix A.3, we discuss the case when the target distribution is weakly log concave (under a bounded fourth moment or bounded covariance matrix assumption) or a perturbation of log-concave distribution. See Corollary 7 for specific details where we provide explicit expressions for the rates that appear in third and fourth columns of Table 3.1.

3.3.3 Mixing time bounds via conductance profile

In this section, we discuss the general results that form the basis of the analysis in this chapter. A standard approach to controlling mixing times is via worst-case conductance bounds. This method was introduced by Jerrum and Sinclair [89] for discrete space chains and then extended to the continuous space settings by Lovász and Simonovits [118], and has been thoroughly studied. See the survey [193] and the references therein for a detailed discussion of conductance based methods for continuous space Markov chains.

Somewhat more recent work on discrete state chains has introduced more refined methods, including those based on the conductance profile [117], the spectral and conductance profile [71], as well as the evolving set method [137]. Here we extend one of the conductance profile techniques from the paper [71] from discrete state to continuous state chains, albeit with several appropriate modifications suited for the general setting.

We first introduce some background on the conductance profile. Given a Markov chain with transition probability $\Theta : \mathcal{X} \times \mathcal{B}(\mathcal{X}) \to \mathbb{R}$, its stationary *flow* $\phi : \mathcal{B}(\mathcal{X}) \to \mathbb{R}$ is defined as

$$\phi(S) = \int_{x \in S} \Theta(x, S^c)\pi^*(x)dx \quad \text{for any } S \in \mathcal{B}(\mathcal{X}). \tag{3.9}$$

Given a set $\Omega \subset \mathcal{X}$, the Ω-*restricted conductance profile* is given by is given by

$$\Phi_\Omega(v) = \inf_{\Pi^*(S \cap \Omega) \in (0,v]} \frac{\phi(S)}{\Pi^*(S \cap \Omega)} \qquad \text{for any } v \in \big(0,\ \Pi^*(\Omega)/2\big]. \tag{3.10}$$

(The classical conductance constant Φ is a special case; it can be expressed as $\Phi = \Phi_{\mathcal{X}}(\frac{1}{2})$.) Moreover, we define the *truncated extension* $\tilde{\Phi}_\Omega$ of the function Φ_Ω to the positive real line as

$$\tilde{\Phi}_\Omega(v) = \begin{cases} \Phi_\Omega(v), & v \in \left(0, \frac{\Pi^*(\Omega)}{2}\right] \\ \Phi_\Omega(\Pi^*(\Omega)/2), & v \in \left[\frac{\Pi^*(\Omega)}{2}, \infty\right). \end{cases} \tag{3.11}$$

In our proofs we use the conductance profile with a suitably chosen set Ω.

Smooth chain assumption: We say that the Markov chain satisfies the *smooth chain assumption* if its transition probability function $\Theta : \mathcal{X} \times \mathcal{B}(\mathcal{X}) \to \mathbb{R}_+$ can be expressed in the form

$$\Theta(x, dy) = \theta(x, y)dy + \alpha_x \delta_x(dy) \quad \text{for all } x, y \in \mathcal{X}, \tag{3.12}$$

where θ is the transition kernel satisfying $\theta(x, y) \geq 0$ for all $x, y \in \mathcal{X}$. Here δ_x denotes the Dirac-delta function at x and consequently, α_x denotes the one-step probability of the chain to stay at its current state x. Note that the three algorithms discussed in this chapter (MRW, MALA and HMC) all satisfy the smooth chain assumption (3.12). Throughout the chapter, when dealing with a general Markov chain, we assume that it satisfies the smooth chain assumption.

Mixing time via conductance profile: We now state our Lemma 4 that provides a control on the mixing time of a Markov chain with continuous-state space in terms of its restricted conductance profile. We show that this control (based on conductance profile) allows us to have a better initialization dependency than the usual conductance based control (see [119, 118, 58]). This method for sharpening the dependence is known

for discrete-state Markov chains; to the best of our knowledge, the following lemma is the first statement and proof of an analogous sharpening for continuous state space chains:

Lemma 4. *Consider a reversible, irreducible, ζ-lazy and smooth Markov chain (3.12) with stationary distribution Π^*. Then for any error tolerance ϵ, and a ϖ-warm distribution μ_0, given a set Ω such that $\Pi^*(\Omega) \geq 1 - \frac{\epsilon^2}{2\varpi^2}$, the ϵ-$\mathcal{L}_2$ mixing time of the chain is bounded as*

$$\tau_2(\epsilon; \mu_0) \leq \int_{4/\varpi}^{8/\epsilon^2} \frac{4\, dv}{\zeta \cdot v \tilde{\Phi}^2_\Omega(v)}, \tag{3.13}$$

where $\tilde{\Phi}_\Omega$ denotes the truncated Ω-restricted conductance profile (3.11).

See Appendix A.1.1 for the proof, which is based on an appropriate generalization of the ideas used by Goel et al. [71] for discrete state chains.

The standard conductance based analysis makes use of the worst-case conductance bound for the chain. In contrast, Lemma 4 relates the mixing time to the conductance profile, which can be seen as point-wise conductance. We use the Ω-restricted conductance profile to state our bounds, because often a Markov chain has poor conductance only in regions that have very small probability under the target distribution. Such a behavior is not disastrous as it does not really affect the mixing of the chain up to a suitable tolerance. Given the bound (3.13), we can derive mixing time bound for a Markov chain readily if we have a bound on the Ω-restricted conductance profile Φ_Ω for a suitable Ω. More precisely, if the Ω-restricted conductance profile Φ_Ω of the Markov chain is bounded as

$$\Phi_\Omega(v) \geq \sqrt{B \log\left(\frac{1}{v}\right)} \quad \text{for } v \in \left[\frac{4}{\varpi}, \frac{1}{2}\right],$$

for some $\varpi > 0$ and Ω such that $\Pi^*(\Omega) \geq 1 - \frac{\epsilon^2}{2\varpi^2}$. Then with a ϖ-warm start, Lemma 4 implies the following useful bound on the mixing time of the ζ-lazy Markov chain:

$$\tau_2(\epsilon; \mu_0) \leq \frac{32}{\zeta B} \log\left(\frac{\log \varpi}{2\epsilon}\right). \tag{3.14}$$

We now relate our result to prior work based on conductance profile.

Prior work: For discrete state chains, a result similar to Lemma 4 was already proposed by Lovász and Kannan (Theorem 2.3 in the paper [117]). Later on, Morris and Perres [137] and Goel et al. [71] used the notion of evolving sets and spectral profile respectively to sharpen the mixing time bounds based on average conductance for discrete-state space chains. In the context of continuous state space chains, Lovász

and Kannan claimed in their original paper [117] that a similar result should hold for general state space chain as well, although we were unable to find any proof of such a general result in that or any subsequent work. Nonetheless, in a later work an average conductance based bound was used by Kannan et al. to derive faster mixing time guarantees for uniform sampling on bounded convex sets for ball walk (see Section 4.3 in the paper [94]). Their proof technique is not easily extendable to more general distributions including the general log-concave distributions in $\mathbb{R}^d$. Instead, our proof of Lemma 4 for general state space chains proceeds by an appropriate generalization of the ideas based on the spectral profile by Goel et al. [71] (for discrete state chains).

Lower bound on conductance profile: Given the bound (3.14), it suffices to derive a lower bound on the conductance profile Φ_Ω of the Markov chain with a suitable choice of the set Ω. We now state a lower bound for the restricted-conductance profile of a general state space Markov chain that comes in handy for this task. We note that a closely related logarithmic-Cheeger inequality was used for sampling from uniform distribution of a convex body [94] and for sampling from log-concave distributions [111] without explicit constants. Since we would like to derive a non-asymptotic mixing rate, we re-derive an explicit form of their result.

Let scalars $s \in (0, 1/2]$, $\omega \in (0, 1)$ and $\Delta > 0$ be given and let $\mathcal{T}_x$ denote the one-step transition distribution of the Markov chain at point x. Suppose that that chain satisfies

$$d_{\mathrm{TV}}\left(\mathcal{T}_x, \mathcal{T}_y\right) \leq 1 - \omega \qquad \text{whenever } x, y \in \Omega \text{ and } \|x - y\|_2 \leq \Delta. \tag{3.15}$$

Lemma 5. *For a given target distribution Π^*, let Ω be a convex measurable set such that the distribution Π^*_Ω satisfies the isoperimetry (or log-isoperimetry) condition (3.6d) with $\mathfrak{a} = 0$ (or $\mathfrak{a} = \frac{1}{2}$ respectively). Then for any Markov chain satisfying the condition (3.15), we have*

$$\Phi_\Omega(v) \geq \frac{\omega}{4} \cdot \min\left\{1, \frac{\Delta}{16\psi_{\mathfrak{a}}} \cdot \log^{\mathfrak{a}}\left(1 + \frac{1}{v}\right)\right\}, \quad \text{for any } v \in \left[0, \frac{\Pi^*(\Omega)}{2}\right]. \tag{3.16}$$

See Appendix A.1.2 for the proof; the extra logarithmic term comes from the logarithmic isoperimetric inequality ($\mathfrak{a} = \frac{1}{2}$).

Faster mixing time bounds: For any target distribution satisfying a logarithmic isoperimetric inequality (including the case of a strongly log-concave distribution), Lemma 5 is a strict improvement of the conductance bounds derived in previous works [115, 58]. Given this result, suppose that we can find a convex set Ω such that $\Pi^*(\Omega) \approx 1$ and the conditions of Lemma 5 are met, then with a ϖ-warm start μ_0, a direct application of the bound (3.14) along with Lemma 5 implies the following bound:

$$\tau_2(\epsilon; \mu_0) \leq O\left(\frac{1}{\omega^2 \Delta^2} \log \frac{\log \varpi}{\epsilon}\right). \tag{3.17}$$

Results known from previous work for continuous state Markov chains scale like $\frac{\log(\varpi/\epsilon)}{\omega^2\Delta^2}$; for instance, see Lemma 6 in the paper [35]. In contrast, the bound (3.17) provides an additional logarithmic factor improvement in the factor ϖ. Such an improvement also allows us to derive a sharper dependency on dimension d for the mixing time for sampling algorithms other than HMC as we now illustrate in the next section.

3.3.4 Improved warmness dependency for MALA and MRW

As discussed earlier, the bound (3.17) helps derive a $\frac{\log\log\varpi}{\log\varpi}$ factor improvement in the mixing time bound from a ϖ-warm start in comparison to earlier conductance based results. In many settings, a suitable choice of initial distribution has a warmness parameter that scales exponentially with dimension d, e.g., $\varpi = O(e^d)$. For such cases, this improvement implies a gain of $O(\frac{d}{\log d})$ in mixing time bounds. As already noted the distribution $\mu_\dagger = \mathcal{N}(x^*, \frac{1}{L}\mathbb{I}_d)$ is a feasible starting distribution[7] whose warmness scales exponentially with dimension d. We now state sharper mixing time bounds for MALA and MRW with $\mu_\dagger$ as the starting distribution. In the result, we use c_1, c_2 to denote positive universal constants.

Theorem 4. *Assume that the target distribution Π^* satisfies the conditions (3.6a) and (3.6b) (i.e. that the negative log-density is L-smooth and m-strongly convex). Then given the initial distribution $\mu_\dagger = \mathcal{N}(x^*, \frac{1}{L}\mathbb{I}_d)$, the $\frac{1}{2}$-lazy versions of MRW and MALA (Algorithms 1 and 2) with step sizes*

$$\eta_{MRW} = c_2 \cdot \frac{1}{Ld\kappa}, \quad \text{and} \quad \eta_{MALA} = c_1 \cdot \frac{1}{Ld \cdot \max\left\{1, \sqrt{\kappa/d}\right\}} \tag{3.18}$$

respectively, satisfy the mixing time bounds

$$\tau_2^{MRW}(\epsilon; \mu_0) = O\left(d\kappa^2 \log\frac{d}{\epsilon}\right), \quad \text{and} \tag{3.19a}$$

$$\tau_2^{MALA}(\epsilon; \mu_0) = O\left(d\kappa \log\frac{d}{\epsilon} \cdot \max\left\{1, \sqrt{\frac{\kappa}{d}}\right\}\right). \tag{3.19b}$$

The proof is omitted as it directly follows from the conductance profile based mixing time bound in Lemma 4, Lemma 5 and the overlap bounds for MALA and MRW provided in [58]. Theorem 4 states that the mixing time bounds for MALA and MRW with the feasible distribution $\mu_\dagger$ as the initial distribution scale as $\widetilde{O}(d\kappa \log(1/\epsilon))$ and $\widetilde{O}(d\kappa^2 \log(1/\epsilon))$. Once again, we note that in light of the inequality (2.5b) we obtain same bounds for the number of steps taken by these algorithms to mix within ϵ total-variation distance of the target distribution Π^*. Consequently, our results improve upon the previously known mixing time bounds for MALA and MRW [58] for

[7] See Section 3.2 of the paper [58], where the authors show that computing x^* is not expensive and even approximate estimates of x^* and L are sufficient to provide a feasible starting distribution.

strongly log-concave distributions. With $\mu_\dagger$ as the initial distribution, the authors had derived bounds of order $\widetilde{O}(d^2\kappa \log(1/\epsilon))$ and $\widetilde{O}(d^2\kappa^2 \log(1/\epsilon))$ for MALA and MRW respectively (cf. Corollary 1 [58]). However, the authors stated that their numerical experiments suggested a better dependency on the dimension for the mixing time. Indeed the mixing time bounds from Theorem 4 are smaller by a factor of $\frac{d}{\log d}$, compared to their bounds for both of these chains thereby resolving their open question. Nonetheless, it is still an open question how to establish a lower bound on the mixing time of these sampling algorithms.

3.4 Numerical experiments

In this section, we numerically compare HMC with MALA and MRW to verify that our suggested step-size and leapfrog steps lead to faster convergence for the HMC algorithm. We adopt the step-size choices for MALA and MRW given in Dwivedi et al. [58], whereas the choices for stepsize and leapfrog rounds for HMC are taken from Corollary 6 in this chapter.

In this simulation, we check the dimension d dependency and condition number κ dependency in the multivariate Gaussian case under our step-size choices. We consider sampling from the multivariate Gaussian distribution with density

$$\Pi^*(x) \propto e^{-\frac{1}{2}x^\top \Sigma^{-1} x}, \tag{3.20}$$

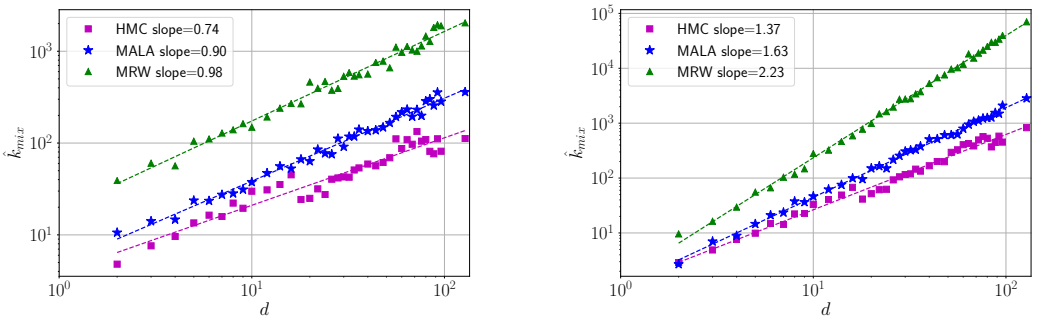
for some covariance matrix $\Sigma \in \mathbb{R}^{d\times d}$. The log density (disregarding constants) and its derivatives are given by

$$f(x) = \frac{1}{2}x^\top \Sigma^{-1} x, \quad \nabla f(x) = \Sigma^{-1}x, \quad \text{and} \quad \nabla^2 f(x) = \Sigma^{-1}.$$

Consequently, the function f is strongly convex with parameter $m = 1/\lambda_{\max}(\Sigma)$ and smooth with parameter $L = 1/\lambda_{\min}(\Sigma)$. For convergence diagnostics, we use the error in quantiles along different directions. Using the exact quantile information for each direction for Gaussians, we measure the error in the 75% quantile of the sample distribution and the true distribution in the *least favorable direction*, i.e., along the eigenvector of Σ corresponding to the eigenvalue $\lambda_{\max}(\Sigma)$. The approximate mixing time is defined as the smallest iteration when this error falls below δ. We use $\mu_0 = \mathcal{N}(0, L^{-1}\mathbb{I}_d)$ as the initial distribution.

(a) Dimension dependency for fixed κ: For a condition number $\kappa = 4$, we vary dimension d from 2 to 128. The parameters for HMC-(K, η) are chosen according to the warm start case in Corollary 3, and for MRW and MALA are chosen according to the paper [58]. We simulate 10 independent runs of the three chains each with 100 samples to determine the approximate mixing time. The final approximate mixing time for each walk is the average of that over these 10 independent runs. Figure 3.1 (a) shows

the dependency of the approximate mixing time as a function of dimension d for the three random walks in log-log scale. To examine the dimension dependency, we perform linear regression for approximate mixing time with respect to dimensions in the log-log scale. The least-squares fits of the slopes for HMC, MALA and MRW are 0.74(±0.22), 0.90(±0.11) and 0.98(±0.14), respectively. Standard errors of the regression coefficient is reported in parentheses. The corresponding theoretical slopes (seen from Table 3.2) are $0.67, 1.0, 1.0$ respectively.



(a) $\kappa = 4$ (b) $\kappa = d^{\frac{2}{3}}$

Figure 3.1. Approximate mixing time using discrete TV error as a function of dimension on Gaussian density (3.20) where the covariance has a condition number κ that is (a) constant 4 and (b) scales with dimension d. Please see the main text for further discussion.

(b) Dimension dependency for $\kappa = d^{2/3}$: For this set of simulations, we vary the dimension d from 2 to 128, and in all cases, construct a problem with condition number $\kappa = d^{2/3}$. The step η and number of leapfrog updates K are chosen as in the second row of Table A.1 given in Appendix A.4. We simulated 10 independent runs of the three chains each with 1000 samples to determine the approximate mixing time. The final approximate mixing time for each walk is the averaged time across these 10 independent runs. Figure 3.1 (b) shows the dependency of the approximate mixing time as a function of dimension d for the three random walks in log-log scale. In order to estimate the exponent α in the dimension dependency d^{α}, we perform a linear regression of the log mixing time on the log dimension; doing so yields estimated exponents $\widehat{\alpha}$ of 1.37(±0.18), 1.63(±0.10) and 2.23(±0.12) for HMC, MALA and MRW, respectively. Standard errors of the regression coefficient is reported in parentheses. The theoretical guarantees given in Table A.2 (in Appendix A.4) correspond to the exponents of 1.58, 1.67 and 2.33 for the three algorithms respectively.

3.5 Proofs

This section is devoted primarily to the proof of Theorem 3. In order to do so, we begin with the mixing time bound based on the conductance profile from Lemma 4. We then seek to apply Lemma 5 in order derive a bound on the conductance profile itself. However, in order to do so, we need to derive bound on the overlap between the proposal distributions of HMC at two nearby points and show that the Metropolis-Hastings step only modifies the proposal distribution by a relatively small amount. This control is provided by Lemma 6, stated in Section 3.5.1. We use it to prove Theorem 3 in Section 3.5.2. Finally, Section 3.5.3 is devoted to the proof of Lemma 6.

3.5.1 Overlap bounds for HMC

In this subsection, we derive two important bounds for the Metropolized HMC chain: (1) first, we quantify the overlap between proposal distributions of the chain for nearby points, and, (2) second, we show that the distortion in the proposal distribution introduced by the Metropolis-Hastings accept-reject step can be controlled if an appropriate step-size is chosen. Putting the two pieces together enables us to invoke Lemma 5 to prove Theorem 3.

In order to do so, we begin with some notation. Let $\mathcal{T}$ denote the transition operator of the HMC chain with leapfrog integrator taking step-size η and number of leapfrog updates K. Let $\mathcal{P}_x$ denote the proposal distribution at $x \in \mathcal{X}$ for the chain before the accept-reject step and the lazy step. Let $\mathcal{T}_x^{\text{before-lazy}}$ denote the corresponding transition distribution after the proposal and the accept-reject step, before the lazy step. By definition, we have

$$\mathcal{T}_x(A) = \zeta \delta_x(A) + (1-\zeta)\mathcal{T}_x^{\text{before-lazy}}(A) \qquad \text{for any measurable set } A \in \mathcal{B}(\mathcal{X}). \tag{3.21}$$

Our proofs make use of the Euclidean ball $\mathcal{R}_s$ defined in equation (3.25). At a high level, the HMC chain has bounded gradient inside the ball $\mathcal{R}_s$ for a suitable choice of s, and the gradient of the log-density gets too large outside such a ball making the chain unstable in that region. However, since the target distribution has low mass in that region, the chain's visit to the region outside the ball is a rare event and thus we can focus on the chain's behavior inside the ball to analyze its mixing time.

In the next lemma, we state the overlap bounds for the transition distributions of the HMC chain. For a fixed univeral constant c, we require

$$K^2\eta^2 \leq \frac{1}{4\max\left\{d^{\frac{1}{2}}L, d^{\frac{2}{3}}L_{\text{H}}^{\frac{2}{3}}\right\}}, \qquad \text{and} \tag{3.22a}$$

$$\eta^2 \leq \frac{1}{cL}\min\left\{\frac{1}{K^2}, \frac{1}{Kd^{\frac{1}{2}}}, \frac{1}{K^{\frac{2}{3}}d^{\frac{1}{3}}\left(\frac{M^2}{L}\right)^{\frac{1}{3}}}, \frac{1}{K\frac{M}{L^{\frac{1}{2}}}}, \frac{1}{K^{\frac{2}{3}}d}\frac{L}{L_{\text{H}}^{\frac{2}{3}}}, \frac{1}{K^{\frac{4}{3}}\frac{M}{L^{\frac{1}{2}}}}\left(\frac{L}{L_{\text{H}}^{\frac{2}{3}}}\right)^{\frac{1}{2}}\right\}. \tag{3.22b}$$

Lemma 6. *Consider a $(L, L_H, s, \psi_{\mathfrak{a}}, M)$-regular target distribution (cf. Assumption (A)) with Ω the convex measurable set satisfying (3.6e). Then with the parameters (K, η) satisfying $K\eta \leq \frac{1}{4L}$ and condition (3.22a), the HMC-(K, η) chain satisfies*

$$\sup_{\|q_0 - \tilde{q}_0\|_2 \leq \frac{K\eta}{4}} d_{TV}(\mathcal{P}_{q_0}, \mathcal{P}_{\tilde{q}_0}) \leq \frac{1}{2}. \tag{3.23a}$$

If, in addition, condition (3.22b) holds, then we have

$$\sup_{x \in \Omega} d_{TV}\left(\mathcal{P}_x, \mathcal{T}_x^{\textit{before-lazy}}\right) \leq \frac{1}{8}. \tag{3.23b}$$

See Section 3.5.3 for the proof.

Lemma 6 is crucial to the analysis of HMC as it enables us to apply the conductance profile based bounds discussed in Section 3.3.3. It reveals two important properties of the Metropolized HMC. First, from equation (3.23a), we see that proposal distributions of HMC at two different points are close if the two points are close. This is proved by controlling the KL-divergence of the two proposal distributions of HMC via change of variable formula. Second, equation (3.23b) shows that the accept-reject step of HMC is well behaved inside Ω provided the gradient is bounded by M.

3.5.2 Proof of Theorem 3

We are now equipped to prove our main theorem. In order to prove Theorem 3, we begin by using Lemma 5 and Lemma 6 to derive an explicit bound for on the HMC conductance profile. Given the assumptions of Theorem 3, conditions (3.22a) and (3.22b) hold, enabling us to invoke Lemma 6 in the proof.

Define the function $\Psi_\Omega : [0, 1] \mapsto \mathbb{R}_+$ as

$$\Psi_\Omega(v) = \begin{cases} \frac{1}{32} \cdot \min\left\{1, \frac{K\eta}{64\psi_{\mathfrak{a}}} \log^{\mathfrak{a}}\left(\frac{1}{v}\right)\right\} & \text{if } v \in \left[0, \frac{1-s}{2}\right]. \\ \frac{K\eta}{2048\psi_{\mathfrak{a}}}, & \text{if } v \in \left(\frac{1-s}{2}, 1\right]. \end{cases} \tag{3.24}$$

This function acts as a lower bound on the truncated conductance profile. Define the Euclidean ball

$$\mathcal{R}_s = \mathbb{B}\left(x^\star, r(s)\sqrt{\frac{d}{m}}\right), \tag{3.25}$$

and consider a pair $(x, y) \in \mathcal{R}_s$ such that $\|x - y\|_2 \leq \frac{1}{4}K\eta$. Invoking the decomposition (3.21) and applying triangle inequality for ζ-lazy HMC, we have

$$\begin{aligned} d_{\mathrm{TV}}\left(\mathcal{T}_x, \mathcal{T}_y\right) &\leq \zeta + (1-\zeta)\, d_{\mathrm{TV}}\left(\mathcal{T}_x^{\text{before-lazy}}, \mathcal{T}_y^{\text{before-lazy}}\right) \\ &\leq \zeta + (1-\zeta)\left(d_{\mathrm{TV}}\left(\mathcal{T}_x^{\text{before-lazy}}, \mathcal{P}_y\right) + d_{\mathrm{TV}}\left(\mathcal{P}_x, \mathcal{P}_y\right) + d_{\mathrm{TV}}\left(\mathcal{P}_x, \mathcal{T}_y^{\text{before-lazy}}\right)\right) \\ &\overset{(i)}{\leq} \zeta + (1-\zeta)\left(\frac{1}{4} + \frac{1}{2} + \frac{1}{4}\right) \\ &= 1 - \frac{1-\zeta}{4}, \end{aligned}$$

where step (i) follows from the bounds (3.23a) and (3.23b) from Lemma 6. For $\zeta = \frac{1}{2}$, substituting $\omega = \frac{1}{8}$, $\Delta = \frac{1}{4}K\eta$ and the convex set $\Omega = \mathcal{R}_s$ into Lemma 5, we obtain that

$$\Phi_\Omega(v) \geq \frac{1}{32} \cdot \min\left\{1, \frac{K\eta}{64\psi_{\mathfrak{a}}} \log^{\mathfrak{a}}\left(1 + \frac{1}{v}\right)\right\}, \quad \text{for } v \in \left[0, \frac{1-s}{2}\right].$$

Here $\mathfrak{a}$ equals to $\frac{1}{2}$ or 0, depending on the assumption (3.6d). By the definition of the truncated conductance profile (3.11), we have that $\tilde{\Phi}_\Omega(v) \geq \frac{K\eta}{2048\psi_{\mathfrak{a}}}$ for $v \in \left[\frac{1-s}{2}, 1\right]$. As a consequence, Ψ_Ω is effectively a lower bound on the truncated conductance profile. Note that the assumption (A) ensures the existence of Ω such that $\Pi^*(\Omega) \geq 1 - s$ for $s = \frac{\epsilon^2}{2\varpi^2}$. Putting the pieces together and applying Lemma 4 with the convex set Ω concludes the proof of the theorem.

3.5.3 Proof of Lemma 6

In this subsection, we prove the two main claims (3.23a) and (3.23b) in Lemma 6. Before going into the claims, we first provide several convenient properties about the HMC proposal.

Properties of the HMC proposal

Recall the Hamiltonian Monte Carlo (HMC) with leapfrog integrator (3.4c). Using an induction argument, we find that the final states in one iteration of K steps of the HMC chain, denoted by q_K and p_K satisfy

$$p_K = p_0 - \frac{\eta}{2}\nabla f(q_0) - \sum_{j=1}^{K-1} \nabla f(q_j) - \frac{\eta}{2}\nabla f(q_K), \tag{3.26a}$$

$$\text{and} \quad q_K = q_0 + K\eta p_0 - \frac{K\eta^2}{2}\nabla f(q_0) - \eta^2 \sum_{j=1}^{K-1} (K-j)\, \nabla f(q_j). \tag{3.26b}$$

It is easy to see that for $k \in [K]$, q_k can be seen as a function of the initial state q_0 and p_0. We denote this function as the *forward mapping* F,

$$q_k =: F_k(p_0, q_0) \quad \text{and} \quad q_K =: F_K(p_0, q_0) =: F\left(p_0, q_0\right) \tag{3.26c}$$

where we introduced the simpler notation $F := F_K$ for the final iterate. The forward mappings F_k and F are deterministic functions that only depends on the gradient ∇f, the number of leapfrog updates K and the step size η.

Denote $\mathbf{J}_x F$ as the Jacobian matrix of the forward mapping F with respect to the first variable. By definition, it satisfies

$$\left[\mathbf{J}_x F(x, q_0)\right]_{ij} = \frac{\partial}{\partial x_j} \left[F(x, q_0)\right]_i, \quad \text{for all} \quad i, j \in [d]. \tag{3.26d}$$

Similarly, denote $\mathbf{J}_y F$ as the Jacobian matrix of the forward mapping F with respect to the second variable. The following lemma characterizes the eigenvalues of the Jacobian $\mathbf{J}_x F$.

Lemma 7. *Suppose the log density f is L-smooth. For the number of leapfrog steps and step-size satisfying $K^2\eta^2 \leq \frac{1}{4L}$, we have*

$$\left\| K\eta \mathbb{I}_d - \mathbf{J}_x F(x, y) \right\|_2 \leq \frac{1}{8} K\eta, \quad \textit{for all} \quad x, y \in \mathcal{X} \textit{ and } i \in [d].$$

Also all eigenvalues of $\mathbf{J}_x F(x, y)$ have absolute value greater or equal to $\frac{7}{8} K\eta$.

See Appendix A.1.3 for the proof.

Since the Jacobian is invertible for $K^2\eta^2 \leq \frac{1}{4L}$, we can define the inverse function of F with respect to the first variable as the backward mapping G. We have

$$F\left(G(x, y), y\right) = x, \quad \text{for all} \quad x, y \in \mathcal{X}. \tag{3.27}$$

Moreover as a direct consequence of Lemma 7, we obtain that the magnitude of the eigenvalues of the Jacobian matrix $\mathbf{J}_x G(x, y)$ lies in the interval $\left[\frac{8}{9K\eta}, \frac{8}{7K\eta}\right]$. In the next lemma, we state another set of bounds on different Jacobian matrices:

Lemma 8. *Suppose the log density f is L-smooth. For the number of leapfrog steps and step-size satisfying $K^2\eta^2 \leq \frac{1}{4L}$, we have*

$$\left\| \mathbf{J}_y G(x, y) \right\|_2 \leq \frac{4}{3K\eta}, \quad \textit{for all} \quad x, y \in \mathcal{X}, \quad \textit{and} \tag{3.28a}$$

$$\left\| \frac{\partial F_k(G(x, y), y)}{\partial y} \right\|_2 \leq 3, \qquad \textit{for all} \quad k \in [K]. \tag{3.28b}$$

See Appendix A.1.3 for the proof.

Next, we would like to obtain a bound on the quantity $\frac{\partial \log\det \mathbf{J}_x G(x, q_0)}{\partial y}$. Applying the chain rule, we find that

$$\frac{\partial \log\det \mathbf{J}_x G(x, q_0)}{\partial y} = \begin{bmatrix} \operatorname{trace}\left([\mathbf{J}_x G(x, q_0)]^{-1} \mathbf{J}_{xy_1} G(x, q_0)\right) \\ \vdots \\ \operatorname{trace}\left([\mathbf{J}_x G(x, q_0)]^{-1} \mathbf{J}_{xy_d} G(x, q_0)\right) \end{bmatrix}. \tag{3.29}$$

Here $\mathbf{J}_{xy}G(x,q_0)$ is a third order tensor and we use $\mathbf{J}_{xy_l}G(x,q_0)$ to denote the matrix corresponding to the l-th slice of the tensor which satisfies

$$\left[\mathbf{J}_{xy_l}G(x,q_0)\right]_{ij} = \frac{\partial\partial}{\partial x_j y_l}\left[F(x,q_0)\right]_i, \qquad \text{for all} \quad i,j,l \in [d].$$

Lemma 9. *Suppose the log density f is L-smooth and L_H-Hessian Lipschitz. For the number of leapfrog steps and step-size satisfying $K^2\eta^2 \leq \frac{1}{4L}$, we have*

$$\left\|\frac{\partial \log\det \mathbf{J}_x G(x,q_0)}{\partial y}\right\|_2 = \left\|\begin{bmatrix} trace\left([\mathbf{J}_x G(x,q_0)]^{-1}\mathbf{J}_{xy_1}G(x,q_0)\right) \\ \vdots \\ trace\left([\mathbf{J}_x G(x,q_0)]^{-1}\mathbf{J}_{xy_d}G(x,q_0)\right)\end{bmatrix}\right\|_2 \leq 2dK^2\eta^2 L_H.$$

See Appendix A.1.3 for the proof.

As a direct consequence of the equation (3.26b) at k-th step of leapfrog updates, we obtain the following two bounds for the difference between successive F_k terms that come in handy later in our proofs.

Lemma 10. *Suppose that the log density f is L-smooth. For the number of leapfrog steps and step-size satisfying $K^2\eta^2 \leq \frac{1}{4L}$, we have*

$$\|F_k(p_0,q_0) - q_0\|_2 \leq 2k\eta\|p_0\|_2 + 2k^2\eta^2\|\nabla f(q_0)\|_2 \quad \text{for } k \in [K], \text{ and} \tag{3.30a}$$

$$\|F_{k+1}(p_0,q_0) - F_k(p_0,q_0)\|_2 \leq 2\eta\|p_0\|_2 + 2(k+1)\eta^2\|\nabla f(q_0)\|_2 \text{ for } k \in [K-1]. \tag{3.30b}$$

See Appendix A.1.3 for the proof.

We now turn to the proof the two claims in Lemma 6. Note that the claim (3.23a) states that the proposal distributions at two close points are close; the claim (3.23b) states that the proposal distribution and the transition distribution are close.

Proof of claim (3.23a) in Lemma 6

In order to bound the distance between proposal distributions of nearby points, we prove the following stronger claim: For a L-smooth L_{H}-Hessian-Lipschitz target distribution, the proposal distribution of the HMC algorithm with step size η and leapfrog steps K such that $K\eta \leq \frac{1}{4L}$ satisfies

$$d_{\mathrm{TV}}\left(\mathcal{P}_{q_0}, \mathcal{P}_{\tilde{q}_0}\right) \leq \left(\frac{2\|q_0-\tilde{q}_0\|_2^2}{K^2\eta^2} + 3\sqrt{d}K\eta L\|q_0-\tilde{q}_0\|_2 + 4dK^2\eta^2 L_{\mathrm{H}}\|q_0-\tilde{q}_0\|_2\right)^{1/2}, \tag{3.31}$$

for all $q_0, \tilde{q}_0 \in \mathbb{R}^d$. Then for any two points $q_0, \tilde{q}_0$ such that $\|q_0 - \tilde{q}_0\|_2 \leq \frac{1}{4}K\eta$, under the condition (3.22a), i.e., $K^2\eta^2 \leq \frac{1}{4\max\left\{d^{\frac{1}{2}}L, d^{\frac{2}{3}}L_{\mathrm{H}}^{\frac{2}{3}}\right\}}$, we have

$$d_{\mathrm{TV}}\left(\mathcal{P}_{q_0}, \mathcal{P}_{\tilde{q}_0}\right) \leq \left(\frac{1}{8} + \frac{3}{64} + \frac{1}{64}\right)^{1/2} \leq \frac{1}{2},$$

and the claim (3.23a) follows.

The proof of claim (3.31) involves the following steps: (1) we make use of the update rules (3.26b) and change of variable formula to obtain an expression for the density of q_n in terms of q_0, (2) then we use Pinsker's inequality and derive expressions for the KL-divergence between the two proposal distributions, and (3) finally, we upper bound the KL-divergence between the two distributions using different properties of the forward mapping F from Appendix 3.5.3.

According to the update rule (3.26b), the proposals from two initial points q_0 and $\tilde{q}_0$ satisfy respectively

$$q_K = F(p_0, q_0), \quad \text{and} \quad \tilde{q}_K = F(\tilde{p}_0, \tilde{q}_0),$$

where p_0 and $\tilde{p}_0$ are independent random variable from Gaussian distribution $\mathcal{N}(0, \mathbb{I}_d)$.

Denote ρ_{q_0} as the density function of the proposal distribution $\mathcal{P}_{q_0}$. For two different initial points q_0 and $\tilde{q}_0$, the goal is to bound the total variation distance between the two proposal distribution, which is by definition

$$d_{\mathrm{TV}}\left(\mathcal{P}_{q_0}, \mathcal{P}_{\tilde{q}_0}\right) = \frac{1}{2}\int_{x\in\mathcal{X}} \left|\rho_{q_0}(x) - \rho_{\tilde{q}_0}(x)\right| dx. \tag{3.32}$$

Given q_0 fixed, the random variable q_K can be seen as a transformation of the Gaussian random variable p_0 through the function $F(\cdot, q_0)$. When F is invertible, we can use the change of variable formula to obtain an explicit expression of the density ρ_{q_0}:

$$\rho_{q_0}(x) = \varphi\left(G(x, q_0)\right) \det\left(\mathbf{J}_x G(x, q_0)\right), \tag{3.33}$$

where φ is the density of the standard Gaussian distribution $\mathcal{N}(0, \mathbb{I}_d)$. Note that even though explicit, directly bounding the total variation distance (3.32) using the complicated density expression (3.33) is difficult. We first use Pinsker's inequality [41] to give an upper bound of the total variance distance in terms of KL-divergence

$$d_{\mathrm{TV}}\left(\mathcal{P}_{q_0}, \mathcal{P}_{\tilde{q}_0}\right) \leq \sqrt{2\mathrm{KL}(\mathcal{P}_{q_0} \parallel \mathcal{P}_{\tilde{q}_0})}, \tag{3.34}$$

and then upper bound the KL-divergence. Plugging the density (3.33) into the KL-divergence formula, we obtain that

$$\begin{aligned}
\mathrm{KL}(\mathcal{P}_{q_0} \parallel \mathcal{P}_{\tilde{q}_0}) &= \int_{\mathbb{R}^d} \rho_{q_0}(x) \log\left(\frac{\rho_{q_0}(x)}{\rho_{\tilde{q}_0}(x)}\right) dx \\
&= \int_{\mathbb{R}^d} \rho_{q_0}(x) \left[\log\left(\frac{\varphi\left(G(x,q_0)\right)}{\varphi\left(G(x,\tilde{q}_0)\right)}\right) + \log\det \mathbf{J}_x G(x,q_0) - \log\det \mathbf{J}_x G(x,\tilde{q}_0)\right] dx \\
&= \underbrace{\int_{\mathbb{R}^d} \rho_{q_0}(x) \left[\frac{1}{2}\left(-\left\|G(x,q_0)\right\|_2^2 + \left\|G(x,\tilde{q}_0)\right\|_2^2\right)\right] dx}_{T_1} \\
&\qquad + \underbrace{\int_{\mathbb{R}^d} \rho_{q_0}(x) \left[\log\det \mathbf{J}_x G(x,q_0) - \log\det \mathbf{J}_x G(x,\tilde{q}_0)\right] dx}_{T_2}
\end{aligned} \tag{3.35}$$

We claim the following bounds on the terms T_1 and T_2:

$$|T_1| \leq \frac{8}{9}\frac{\left\|q_0 - \tilde{q}_0\right\|_2^2}{K^2\eta^2} + \frac{3}{2}\sqrt{d}K\eta L \left\|q_0 - \tilde{q}_0\right\|_2, \quad \text{and} \tag{3.36a}$$

$$|T_2| \leq 2dK^2\eta^2 L_{\mathrm{H}} \left\|q_0 - \tilde{q}_0\right\|_2, \tag{3.36b}$$

where the bound on T_2 follows readily from Lemma 9:

$$\begin{aligned}
|T_2| &= \left|\int \rho_{q_0}(x) \left[\log\det \mathbf{J}_x G(x,q_0) - \log\det \mathbf{J}_x G(x,\tilde{q}_0)\right] dx\right| \\
&\leq \left\|\frac{\partial \log\det \mathbf{J}_x G(x,q_0)}{\partial y}\right\|_2 \left\|q_0 - \tilde{q}_0\right\|_2 \\
&\leq 2dK^2\eta^2 L_{\mathrm{H}} \left\|q_0 - \tilde{q}_0\right\|_2.
\end{aligned} \tag{3.37}$$

Putting together the inequalities (3.34), (3.35), (3.36a) and (3.36b) yields the claim (3.31).

It remains to prove the bound (3.36a) on T_1.

Proof of claim (3.36a)**:** For the term T_1, we observe that

$$\frac{1}{2}\left(\left\|G(x,\tilde{q}_0)\right\|_2^2 - \left\|G(x,q_0)\right\|_2^2\right) = \frac{1}{2}\left\|G(x,q_0) - G(x,\tilde{q}_0)\right\|_2^2 - \left(G(x,q_0) - G(x,\tilde{q}_0)\right)^\top G(x,q_0).$$

The first term on the RHS can be bounded via the Jacobian of G with respect to the second variable. Applying the bound (3.28a) from Lemma 8, we find that

$$\left\|G(x,q_0) - G(x,\tilde{q}_0)\right\|_2 \leq \left\|\mathbf{J}_y G(x,y)\right\|_2 \left\|q_0 - \tilde{q}_0)\right\|_2 \leq \frac{4}{3K\eta}\left\|q_0 - \tilde{q}_0)\right\|_2. \tag{3.38}$$

For the second part, we claim that there exists a deterministic function C of q_0 and $\tilde{q}_0$ and independent of x, such that

$$\left\|G(x,q_0) - G(x,\tilde{q}_0) - C(q_0,\tilde{q}_0)\right\|_2 \leq \frac{3}{2}K\eta L \left\|q_0 - \tilde{q}_0\right\|_2. \tag{3.39}$$

Assuming the claim (3.39) as given at the moment, we can further decompose the second part of T_1 into two parts:

$$\left(G(x,q_0) - G(x,\tilde{q}_0)\right)^\top G(x,q_0) \tag{3.40}$$
$$= \left(G(x,q_0) - G(x,\tilde{q}_0) - C(q_0,\tilde{q}_0)\right)^\top G(x,q_0) + C(q_0,\tilde{q}_0)^\top G(x,q_0) \tag{3.41}$$

Applying change of variables along with equation (3.33), we find that

$$\int \rho_{q_0}(x) G(x,q_0) dx = \int \varphi(x) x dx = 0.$$

Furthermore, we also have

$$\begin{aligned}\int_{x\in\mathcal{X}} \rho_{q_0}(x) \left\|G(x,q_0)\right\|_2 dx &= \int_{x\in\mathcal{X}} \varphi(x) \left\|x\right\|_2 dx \\ &\overset{(i)}{\leq} \left[\left(\int_{x\in\mathcal{X}} \varphi(x) \left\|x\right\|_2^2 dx\right)\left(\int_{x\in\mathcal{X}} \varphi(x) dx\right)\right]^{1/2} = \sqrt{d},\end{aligned}$$

where step (i) follows from Cauchy-Schwarz's inequality. Combining the inequalities (3.38), (3.39) and (3.40) together, we obtain the following bound on term T_1:

$$\begin{aligned}|T_1| &= \left|\int \rho_{q_0}(x)\left[-\frac{1}{2}\left\|G(x,q_0)\right\|_2^2 + \frac{1}{2}\left\|G(x,\tilde{q}_0)\right\|_2^2\right] dx\right| \\ &\leq \frac{1}{2}\left|\int \rho_{q_0}(x)\left\|G(x,q_0) - G(x,\tilde{q}_0)\right\|_2^2 dx\right| \\ &\qquad + \left|\int \rho_{q_0}(x)\left\|G(x,q_0) - G(x,\tilde{q}_0) - C(q_0,\tilde{q}_0)\right\|_2 \left\|G(x,q_0)\right\|_2 dx\right| \\ &\leq \frac{8}{9}\frac{\left\|q_0 - \tilde{q}_0\right\|_2^2}{K^2\eta^2} + \frac{3}{2}\sqrt{d}K\eta\left\|q_0 - \tilde{q}_0\right\|_2, \end{aligned} \tag{3.42}$$

which yields the claimed bound on T_1.

We now prove our earlier claim (3.39).

Proof of claim (3.39): For any pair of states q_0 and $\tilde{q}_0$, invoking the definition (3.27) of the map $G(x,\cdot)$, we obtain the following implicit equations:

$$x = q_0 + K\eta G(x,q_0) - K\frac{\eta^2}{2}\nabla f(q_0) - \eta^2\sum_{j=1}^{K-1}(K-j)\nabla f(F_j(G(x,q_0),q_0)), \quad \text{and}$$
$$x = \tilde{q}_0 + K\eta G(x,\tilde{q}_0) - K\frac{\eta^2}{2}\nabla f(\tilde{q}_0) - \eta^2\sum_{j=1}^{K-1}(K-j)\nabla f(F_j(G(x,\tilde{q}_0),\tilde{q}_0)).$$

Taking the difference between the two equations above, we obtain

$$G(x, q_0) - G(x, \tilde{q}_0) - \frac{q_0 - \tilde{q}_0}{K\eta} - \frac{\eta}{2}\left(\nabla f(q_0) - \nabla f(\tilde{q}_0)\right) = \frac{\eta^2}{K\eta}\sum_{k=1}^{K-1}(K-j)\left(\nabla f(F_k(G(x, q_0), q_0)) - \nabla f(F_k(G(x, \tilde{q}_0), \tilde{q}_0))\right).$$

Applying L-smoothness of f along with the bound (3.28b) from Lemma 8, we find that

$$\begin{aligned}\left\|\nabla f(F_k(G(x, q_0), q_0)) - \nabla f(F_k(G(x, \tilde{q}_0), \tilde{q}_0))\right\|_2 &\leq L\left\|\frac{\partial F_k(G(x, y), y)}{\partial y}\right\|_2 \|q_0 - \tilde{q}_0\|_2 \\ &\leq 3L\|q_0 - \tilde{q}_0\|_2.\end{aligned}$$

Putting the pieces together, we find that

$$\left\|G(x, q_0) - G(x, \tilde{q}_0) - \frac{q_0 - \tilde{q}_0}{K\eta} - \frac{1}{2}\left(\nabla f(q_0) - \nabla f(\tilde{q}_0)\right)\right\|_2 \leq \frac{3K\eta L}{2}\|q_0 - \tilde{q}_0\|_2,$$

which yields the claim (3.39).

Proof of claim (3.23b) in Lemma 6

We now bound the distance between the one-step proposal distribution $\mathcal{P}_x$ at point x and the one-step transition distribution $\mathcal{T}_x^{\text{before-lazy}}$ at x obtained after performing the accept-reject step (and no lazy step). Using equation (3.26a), we define the forward mapping E for the variable p_K as follows

$$p_K = E(p_0, q_0) := p_0 - \frac{\eta}{2}\nabla f(q_0) - \eta\sum_{j=1}^{K-1}\nabla f(q_j) - \frac{\eta}{2}\nabla f(q_K).$$

Consequently, the probability of staying at x is given by

$$\mathcal{T}_x^{\text{before-lazy}}(\{x\}) = 1 - \int_{\mathcal{X}} \min\left\{1, \frac{\exp(-\mathcal{H}(E(z, x), F(z, x)))}{\exp(-\mathcal{H}(z, x))}\right\}\varphi_x(z)dz,$$

where the Hamiltonian $\mathcal{H}(q, p) = f(q) + \frac{1}{2}\|p\|_2^2$ was defined in equation (3.3). As a result, the TV-distance between the proposal and transition distribution is given by

$$\begin{aligned}d_{\text{TV}}\left(\mathcal{P}_x, \mathcal{T}_x^{\text{before-lazy}}\right) &= 1 - \int_{\mathcal{X}} \min\left\{1, \frac{\exp(-\mathcal{H}(E(z, x), F(z, x)))}{\exp(-\mathcal{H}(z, x))}\right\}\varphi_x(z)dz \\ &= 1 - \mathbb{E}_{z\sim\mathcal{N}(0, \mathbb{I}_d)}\left[\min\left\{1, \frac{\exp(-\mathcal{H}(E(z, x), F(z, x)))}{\exp(-\mathcal{H}(z, x))}\right\}\right]. \quad (3.43)\end{aligned}$$

An application of Markov's inequality yields that

$$\begin{aligned}&\mathbb{E}_{z\sim\mathcal{N}(0,\mathbb{I}_d)}\left[\min\left\{1,\frac{\exp(-\mathcal{H}(E(z,x),F(z,x)))}{\exp(-\mathcal{H}(z,x))}\right\}\right]\\&\quad\geq\alpha\mathbb{P}_{z\sim\mathcal{N}(0,\mathbb{I}_d)}\left[\frac{\exp(-\mathcal{H}(E(z,x),F(z,x)))}{\exp(-\mathcal{H}(z,x))}\geq\alpha\right],\end{aligned}\tag{3.44}$$

for any $\alpha\in(0,1]$. Thus, to bound the distance $d_{\mathrm{TV}}\left(\mathcal{P}_x,\mathcal{T}_x^{\text{before-lazy}}\right)$, it suffices to derive a high probability lower bound on the ratio $\exp(-\mathcal{H}(E(z,x),F(z,x)))/\exp(-\mathcal{H}(z,x))$ when $z\sim\mathcal{N}(0,\mathbb{I}_d)$.

We now derive a lower bound on the following quantity:

$$\exp\left(-f(F(p_0,q_0))+f(q_0)-\frac{1}{2}\left\|E(p_0,q_0)\right\|_2^2+\frac{1}{2}\left\|p_0\right\|_2^2\right),\quad\text{when } p_0\sim\mathcal{N}(0,\mathbb{I}_d).$$

We derive the bounds on the two terms $-f(F(p_0,q_0))+f(q_0)$ and $\|E(p_0,q_0)\|_2^2$ separately.

Observe that

$$f(F(p_0,q_0))-f(q_0)=\sum_{j=0}^{K-1}\left[f(F_{j+1}(p_0,q_0))-f(F_j(p_0,q_0))\right].$$

The intuition is that it is better to apply Taylor expansion on closer points. Applying the third order Taylor expansion and using the smoothness assumptions (3.6a) and (3.6c) for the function f, we obtain

$$f(x)-f(y)\leq\frac{(x-y)^\top}{2}\left(\nabla f(x)+\nabla f(y)\right)+L_{\mathrm{H}}\left\|x-y\right\|_2^3.$$

For the indices $j\in\{0,\dots,K-1\}$, using F_j as the shorthand for $F_j(p_0,q_0)$, we find that

$$\begin{aligned}&f(F_{j+1})-f(F_j)\\&\leq\frac{(F_{j+1}-F_j)^\top}{2}\left(\nabla f(F_{j+1})+\nabla f(F_j)\right)+L_{\mathrm{H}}\left\|F_{j+1}-F_j\right\|_2^3\\&=\frac{1}{2}\eta p_0^\top\left(\nabla f(F_{j+1})+\nabla f(F_j)\right)\\&\quad-\frac{\eta^2}{2}\left[\frac{1}{2}\nabla f(p_0)+\sum_{k=1}^{j}\nabla f(F_k)\right]^\top\left(\nabla f(F_{j+1})+\nabla f(F_j)\right)+L_{\mathrm{H}}\left\|F_{j+1}-F_j\right\|_2^3,\end{aligned}\tag{3.45}$$

where the last equality follows by definition (3.26c) of the operator F_j.

Now to bound the term $E(p_0, q_0)$, we observe that

$$\begin{aligned}\frac{\|E(p_0,q_0)\|_2^2}{2} &= \frac{\left\|p_0 - \frac{\eta}{2}\nabla f(q_0) - \eta\sum_{j=1}^{K-1}\nabla f(F_j) - \frac{\eta}{2}\nabla f(F_K)\right\|_2^2}{2} \\ &= \frac{\|p_0\|_2^2}{2} - \eta p_0^\top\left(\frac{1}{2}\nabla f(q_0) + \sum_{j=1}^{K-1}\nabla f(F_j) + \frac{1}{2}\nabla f(F_K)\right) \\ &\qquad + \frac{\eta^2}{2}\left\|\frac{1}{2}\nabla f(q_0) + \sum_{j=1}^{K-1}\nabla f(F_j) + \frac{1}{2}\nabla f(F_K)\right\|_2^2. \end{aligned} \tag{3.46}$$

Putting the equations (3.45) and (3.46) together leads to cancellation of many gradient terms and we obtain

$$\begin{aligned} &- f(F(p_0,q_0)) + f(q_0) - \frac{1}{2}\|E(p_0,q_0)\|_2^2 + \frac{1}{2}\|p_0\|_2^2 \\ &\geq \frac{\eta^2}{8}\left(\nabla f(q_0) - \nabla f(F_K)\right)^\top\left(\nabla f(q_0) + \nabla f(F_K)\right) - L_{\mathrm{H}}\sum_{j=0}^{K-1}\|F_{j+1} - F_j\|_2^3 \\ &\geq -\frac{\eta^2 L}{4}\|q_0 - F(p_0,q_0)\|_2\|\nabla f(q_0)\|_2 - \frac{\eta^2 L^2}{2}\|q_0 - F(p_0,q_0)\|_2^2 - L_{\mathrm{H}}\sum_{j=0}^{K-1}\|F_{j+1} - F_j\|_2^3 \end{aligned} \tag{3.47}$$

The last inequality uses the smoothness condition (3.6a) for the function f. Plugging the bounds (3.30a) and (3.30b) in equation (3.47), we obtain a lower bound that only depends on $\|p_0\|_2$ and $\|\nabla f(q_0)\|_2$:

$$\begin{aligned}\text{RHS of (3.47)} \geq &-2K^2\eta^4 L^2\|p_0\|_2^2 - 2K\eta^3 L\|p_0\|_2\|\nabla f(q_0)\|_2 - 2K^2\eta^4 L\|\nabla f(q_0)\|_2^2 \\ &-L_{\mathrm{H}}\left(32K\eta^3\|p_0\|_2^3 + 8K^4\eta^6\|\nabla f(q_0)\|_2^3\right).\end{aligned} \tag{3.48}$$

According to assumption (A), we have bounded gradient in the convex set Ω. For any $x \in \Omega$, we have $\|\nabla f(x)\|_2 \leq M$. Standard Chi-squared tail bounds imply that

$$\mathbb{P}\left[\|p_0\|_2^2 \leq d\alpha_1\right] \geq 1 - \frac{1}{16}, \quad \text{for } \alpha_1 = 1 + 2\sqrt{\log(16)} + 2\log(16). \tag{3.49}$$

Plugging the gradient bound and the bound (3.49) into equation (3.48), we conclude that there exists an absolute constant $c \leq 2000$ such that for η^2 satisfying equation (3.22b), namely

$$\eta^2 \leq \frac{1}{cL}\min\left\{\frac{1}{K^2}, \frac{1}{Kd^{\frac{1}{2}}}, \frac{1}{K^{\frac{2}{3}}d^{\frac{1}{3}}\left(\frac{M^2}{L}\right)^{\frac{1}{3}}}, \frac{1}{K\frac{M}{L^{\frac{1}{2}}}}, \frac{1}{K^{\frac{2}{3}}d}\frac{L}{L_{\mathrm{H}}^{\frac{2}{3}}}, \frac{1}{K^{\frac{4}{3}}\frac{M}{L^{\frac{1}{2}}}}\left(\frac{L}{L_{\mathrm{H}}^{\frac{2}{3}}}\right)^{\frac{1}{2}}\right\},$$

we have

$$\mathbb{P}\left[-f(F(p_0,q_0))+f(q_0)-\frac{1}{2}\left\|E(p_0,q_0)\right\|_2^2+\frac{1}{2}\left\|p_0\right\|_2^2\geq -1/16\right]\geq 1-\frac{1}{16}.$$

Plugging this bound in the inequality (3.44) yields that

$$\mathbb{E}_{z\sim\mathcal{N}(0,\mathbb{I}_d)}\left[\min\left\{1,\frac{\exp(-\mathcal{H}(E(z,x),F(z,x)))}{\exp(-\mathcal{H}(z,x))}\right\}\right]\geq 1-\frac{1}{8},$$

which when plugged in equation (3.43) implies that $d_{\mathrm{TV}}\left(\mathcal{P}_x,\mathcal{T}_x^{\text{before-lazy}}\right)\leq 1/8$ for any $x\in\mathcal{R}_s$, as claimed. The proof is now complete.

3.6 Summary

In this chapter, we derived non-asymptotic bounds on mixing time of Metropolized Hamiltonian Monte Carlo for log-concave distributions. Our results show that by choosing appropriate step-size and number of leapfrog steps, we obtain HMC convergence rate which is faster than the current best convergence rate of MALA. This improvement can be seen as the benefit of using multi-step gradients in HMC. An interesting open problem is to determine whether our HMC mixing rate is tight for log-concave sampling under the assumptions made in the chapter.

Even though, we focused on the problem of sampling only from strongly and weakly log-concave distribution, our Theorem 3 applies to general distributions including nearly log-concave distributions as mentioned in Appendix A.3.2. It would be interesting to determine the explicit HMC mixing rate for these distributions. The other main contribution is to improve the warmness dependency in mixing rates of Metropolized algorithms that are proved previously such as MRW and MALA [58]. Our idea is inspired by the techniques used to improve warmness dependency in the literature of discrete-state Markov chains. It is interesting to ask if this warmness dependency can be further improved to prove a convergence sub-linear in d for HMC even for small condition number κ.

Chapter 4

Sampling on polytopes

In this chapter of the thesis, we continue to study the convergence of sampling algorithms but now the state space is constrained to be a polytope. Polytope-constrained sampling requires additional treatment of the sampling algorithm at the boundary of the polytope to ensure that the algorithm do not end up sampling points outside the polytope and the algorithm keeps the correct stationary distribution. In particular, we propose and analyze two new MCMC sampling algorithms, the Vaidya walk and the John walk, for generating samples from the uniform distribution over a polytope. Both random walks are sampling algorithms derived from interior point methods. The former is based on volumetric-logarithmic barrier introduced by Vaidya whereas the latter uses John's ellipsoids. We show that both the Vaidya walk and John walk mix in significantly fewer steps than the logarithmic-barrier based Dikin walk studied in past work.

4.1 Introduction

Sampling from distributions is a core problem in statistics, probability, operations research, and other areas involving stochastic models [68, 22, 159, 77]. Sampling algorithms are a prerequisite for applying Monte Carlo methods to order to approximate expectations and other integrals. Recent decades have witnessed great success of Markov Chain Monte Carlo (MCMC) algorithms; for instance, see the handbook [23] and references therein. These methods are based on constructing a Markov chain whose stationary distribution is equal to the target distribution, and then drawing samples by simulating the chain for a certain number of steps. An advantage of MCMC algorithms is that they only require knowledge of the target density up to a proportionality constant. However, the theoretical understanding of MCMC algorithms used in practice is far from complete. In particular, a general challenge is to bound the *mixing time* of a given MCMC algorithm, meaning the number of iterations—as a function of the error tolerance δ, problem dimension d and other parameters—for the chain to arrive at a distribution within distance δ of the target.

In this chapter, we study a certain class of MCMC algorithms designed for the problem of drawing samples from the uniform distribution over a polytope. The polytope is specified in the form $\mathcal{K} := \{x \in \mathbb{R}^d \mid Ax \leq b\}$, parameterized by the matrix-vector pair $(A, b) \in \mathbb{R}^{n \times d} \times \mathbb{R}^n$. Our goal is to understand the mixing time for obtaining δ-accurate samples, and how it grows as a function of the pair (n, d). The problem of sampling uniformly from a polytope is important in various applications and methodologies. For instance, it underlies various methods for computing randomized approximations to polytope volumes. There is a long line of work on sampling methods being used to obtain randomized approximations to the volumes of polytopes and other convex bodies (see, e.g., [119, 103, 12, 115, 40]).

Polytope sampling is also useful in developing fast randomized algorithms for convex optimization [13] and sampling contingency tables [97], as well as in randomized methods for approximately solving mixed integer convex programs [84, 83]. Sampling from polytopes is also related to simulations of the hard-disk model in statistical physics [98], as well as to simulations of error events for linear programming in communication [62].

Many MCMC algorithms have been studied for sampling from polytopes, and more generally, from convex bodies. Some early examples include the Ball Walk [119] and the hit-and-run algorithm [12, 115], which apply to sampling from general convex bodies. Although these algorithms can be applied to polytopes, they do not exploit any special structure of the problem. In contrast, the Dikin walk introduced by Kannan and Narayanan[97] is specialized to polytopes, and thus can achieve faster convergence rates than generic algorithms. The Dikin walk was the first sampling algorithm based on a connection to interior point methods for solving linear programs. More specifically, as we discuss in detail below, it constructs proposal distributions based on the standard logarithmic barrier for a polytope. In a later paper, Narayanan [141] extended the Dikin walk to general convex sets equipped with self-concordant barriers.

For a polytope defined by n constraints, Kannan and Narayanan [97] proved an upper bound on the mixing time of the Dikin walk that scales linearly with n. In many applications, the number of constraints n can be much larger than the number of variables d. For example, we could imagine one using many hyperplane constraints to approximate complicated convex sets such as sphere or ellipsoid. For such problems, linear dependence on the number of constraints is not desirable. Consequently, it is natural to ask if it is possible to design a sampling algorithm whose mixing time scales in a sub-linear manner with the number of constraints. Our main contribution is to investigate and answer this question in affirmative—in particular, by designing and analyzing two sampling algorithms with provably faster convergence rates than the the Dikin walk while retaining its advantages over the ball walk and the hit-and-run methods.

Our contributions: We introduce and analyze a new random walk, which we refer to as the *Vaidya walk* since it is based on the *volumetric-logarithmic barrier* introduced by [190]. We show that for a polytope in $\mathbb{R}^d$ defined by n-constraints, the

Vaidya walk mixes in $O\left(n^{1/2}d^{3/2}\right)$ steps, whereas the Dikin walk [97] has mixing time bounded as $O\left(nd\right)$. So the Vaidya walk is better in the regime $n \gg d$. We also propose the *John walk*, which is based on *John ellipsoidal algorithm* in optimization. We show that the John walk has a mixing time of $O\left(d^{2.5} \cdot \log^4(n/d)\right)$ and conjecture that a variant of it could achieve $O\left(d^2 \cdot \text{poly-log}(n/d)\right)$ mixing time. We show that when compared to the Dikin walk, the per-iteration computational complexities of the Vaidya walk and the John walk are within a constant factor and a poly-logarithmic in n/d factor respectively. Thus, in the regime $n \gg d$ the overall upper bound on the complexity of generating an approximately uniform sample follows the order Dikin walk $\gg$ Vaidya walk $\gg$ John walk.

The remainder of the chapter is organized as follows. In Section 4.2, we discuss many polynomial-time random walks on convex sets and polytopes, and motivate the starting point for the new random walks. In Section 4.3, we introduce the new random walks and state bounds on their rates of convergence and provide a sketch of the proof in Section 4.3.5. We discuss the computational complexity of the different random walks and demonstrate the contrast between the random walks for several illustrative examples in Section 4.4. We present the proof of the mixing time for the Vaidya walk in Section 4.5 and defer the analysis of the John walk to the appendix. We conclude with possible extensions of our work in Section 4.6.

4.2 Background and problem set-up

In this section, we first review the rates of convergence of existing random walks on convex sets. After introducing several random walks studied in past work, we introduce the Vaidya and John walks studied in this chapter.

4.2.1 Sampling from polytopes

In this chapter, we consider the problem of drawing a sample uniformly from a polytope. Given a full-rank matrix $A \in \mathbb{R}^{n\times d}$ with $n \geq d$, we consider a polytope $\mathcal{K}$ in $\mathbb{R}^d$ of the form

$$\mathcal{K} := \left\{x \in \mathbb{R}^d \mid Ax \leq b\right\}, \tag{4.1}$$

where $b \in \mathbb{R}^n$ is a fixed vector. Since the uniform distribution on the polytope $\mathcal{K}$ is the primary target distribution considered, in the sequel we use Π^* exclusively to denote the uniform distribution on the polytope $\mathcal{K}$. There are various algorithms to sample a vector from the uniform distribution over $\mathcal{K}$, including the ball walk [119] and hit-and-run algorithms [115]. To be clear, these two algorithms apply to the more general problem of sampling from a convex set; when applied to the polytope $\mathcal{K}$, Table 4.1 shows their complexity relative to the Vaidya walk analyzed in this chapter. Most closely related to our work is the Dikin walk proposed by [97], and a more general random

walk on a Riemannian manifold studied by [141]. Both of these random walks, as with the Vaidya and John walks, can be viewed as randomized versions of the interior point methods used to solve linear programs, and more generally convex programs equipped with suitable barrier functions.

In order to motivate the form of the Vaidya and John walks proposed in this chapter, we begin by discussing the ball walk and then the Dikin walk. For the sake of completeness, we end the section with a brief description another popular sampling algorithm Hit-and-run.

Ball walk: The ball walk of [119] is simple to describe: when at a point $x \in \mathcal{K}$, it draws a new point u from a Euclidean ball of radius $r > 0$ centered at x. Here the radius r is a step size parameter in the algorithm. If the proposed point u belongs to the polytope $\mathcal{K}$, then the walk moves to u; otherwise, the walk stays at x. On the one hand, unlike the walks analyzed in this chapter, the ball walk applies to any convex set, but on the other, its mixing time depends on the condition number $\gamma_{\mathcal{K}}$ of the set $\mathcal{K}$, given by

$$\gamma_{\mathcal{K}} = \inf_{R_{\text{in}}, R_{\text{out}} > 0} \Big\{ \frac{R_{\text{out}}}{R_{\text{in}}} \quad | \quad \mathbb{B}(x, R_{\text{in}}) \subseteq \mathcal{K} \subseteq \mathbb{B}(y, R_{\text{out}}) \quad \text{for some } x, y \in \mathcal{K} \Big\}. \tag{4.2}$$

Mixing time of the ball walk has been improved greatly since it was introduced [96, 94, 111]. Nonetheless, as shown in Table 4.1, the mixing time of the ball walk gets slower when the condition of the set is large; for instance, it scales[1] as d^6 for a set with condition number $\gamma_{\mathcal{K}} = d^2$. One approach to tackle bad conditioning is to use rounding as a pre-processing step, where the set is rounded to bring it in a near-isotropic position, i.e., reduce the condition $\gamma_{\mathcal{K}}$ to near-constant before sampling from it. Nonetheless, these algorithms are themselves based on several rounds of sampling algorithms and the current best algorithm by Lovász [122] puts a convex body into approximately isotropic position, i.e., $\mathcal{O}^*(\sqrt{d})$ rounding with a running time of $\mathcal{O}^*(d^4)$ where we have omitted the dependence on log-factors. If one has more information about the structure of the convex set (and not just oracle access as required by the ball walk), one can potentially exploit it to design fast sampling algorithms which are unaffected by the conditioning of the set thereby reducing the need of the (expensive) pre-processing step. One such algorithm is the Dikin walk for polytopes which we describe next.

Dikin walk: The Dikin walk [97] is similar in spirit to the ball walk, except that it proposes a point drawn uniformly from a *state-dependent* ellipsoid known as the Dikin ellipsoid [53, 150]. It then applies an accept-reject step to adjust for the difference in the volumes of these ellipsoids at different states. The state-dependent choice of the ellipsoid allows the Dikin walk to adapt to the boundary structure. A key property of

[1] Although, very recently Lee and Vempala [111] improved the mixing time of the ball walk for isotropic sets which have $\gamma_{\mathcal{K}} = \mathcal{O}(\sqrt{d})$ improved from $O\left(d^3\right)$ to $O\left(d^{2.5}\right)$.

the Dikin ellipsoid of unit radius—in contrast to the Euclidean ball that underlies the ball walk—is that it is always contained within $\mathcal{K}$, as is known from classic results on interior point methods [150]. Furthermore, the Dikin walk is affine invariant, meaning that its behavior does not change under linear transformations of the problem. As a consequence, the Dikin mixing time does not depend on the condition number $\gamma_{\mathcal{K}}$. In a variant of this random walk [141], uniform proposals in the ellipsoid are replaced by Gaussian proposals with covariance specified by the ellipsoid, and it is shown that with high probability, the proposal falls within the polytope.

The Dikin walk is closely related to the interior point methods for solving linear programs. In order to understand the Vaidya and John walks, it is useful to understand this connection in more detail. Suppose that our goal is to optimize a convex function over the polytope $\mathcal{K}$. A barrier method is based on converting this constrained optimization problem to a sequence of unconstrained ones, in particular by using a barrier to enforce the linear constraints defining the polytope. Letting $a_i^\top$ denote the i-th row vector of matrix A, the *logarithmic-barrier* for the polytope $\mathcal{K}$ given by the function

$$\mathcal{F}(x) := -\sum_{i=1}^{n} \log(b_i - a_i^T x). \tag{4.3}$$

For each $i \in [n]$, we define the scalar $s_{x,i} := (b_i - a_i^T x)$, and we refer to the vector $s_x := (s_{x,1}, \ldots, s_{x,n})^\top$ as the *slackness at* x.

Each step of an interior point algorithm [21] involves (approximately) solving a linear system involving the Hessian of the barrier function, which is given by

$$\nabla^2 \mathcal{F}(x) := \sum_{i=1}^{n} \frac{a_i a_i^\top}{s_{x,i}^2}. \tag{4.4}$$

In the Dikin walk [97], given a current iterate x, the algorithm chooses a point uniformly at random from the ellipsoid

$$\{u \in \mathbb{R}^d \mid (u-x)^\top D_x (u-x) \leq R\}, \tag{4.5}$$

where $D_x := \nabla^2 \mathcal{F}(x)$ is the Hessian of the log barrier function, and $R > 0$ is a user-defined radius. In an alternative form of the Dikin walk [141, 173], the proposal vector $u \in \mathbb{R}^d$ is drawn randomly from a Gaussian centered at x, and with covariance equal to a scaled copy of $(D_x)^{-1}$. Note that in contrast to the ball walk, the proposal distribution now depends on the current state.

Vaidya walk: For the *Vaidya walk* analyzed in this chapter, we instead generate proposals from the ellipsoids defined, for each $x \in \text{int}\,(\mathcal{K})$, by the positive definite

matrix

$$V_x := \sum_{i=1}^{n} (\sigma_{x,i} + \beta_{\mathrm{V}}) \frac{a_i a_i^\top}{s_{x,i}^2}, \qquad \text{where} \tag{4.6a}$$

$$\beta_{\mathrm{V}} := d/n \quad \text{and} \quad \sigma_x := \left(\frac{a_1^\top (\nabla^2 \mathcal{F}_x)^{-1} a_1}{s_{x,1}^2}, \ldots, \frac{a_n^\top (\nabla^2 \mathcal{F}_x)^{-1} a_n}{s_{x,n}^2} \right)^\top. \tag{4.6b}$$

The entries of the the vector σ_x are known as the leverage scores associated with the matrix $\nabla^2 \mathcal{F}_x$ (4.4), and are commonly used to measure the importance of rows in a linear system [126]. The matrix V_x is related to the Hessian of the function $x \mapsto \mathcal{V}_x$ given by

$$\mathcal{V}_x := \log\det \nabla^2 \mathcal{F}_x + \beta_{\mathrm{V}} \mathcal{F}_x. \tag{4.7}$$

This particular combination of the *volumetric barrier* and the *logarithmic barrier* was introduced by Vaidya et al. [190, 191] in the context of interior point methods, hence our name for the resulting random walk.

John walk: We now describe the John walk. For any vector $w \in \mathbb{R}^n$, let $W := \mathrm{diag}(w)$ denote the diagonal matrix with $W_{ii} = w_i$ for each $i \in [n]$. Let $S_x = \mathrm{diag}(s_x)$ denote the slackness matrix at x. It is easy to see that S_x is positive semidefinite for all $x \in \mathcal{K}$, and strictly positive definite for all $x \in \mathrm{int}\,(\mathcal{K})$. The (scaled) inverse covariance matrix underlying the John walk is given by

$$J_x := \sum_{i=1}^{n} \zeta_{x,i} \frac{a_i a_i^\top}{s_{x,i}^2}, \tag{4.8}$$

where for each $x \in \mathrm{int}\,(\mathcal{K})$, the weight vector $\zeta_x \in \mathbb{R}^n$ is obtained by solving the convex program

$$\zeta_x := \arg\min_{w \in \mathbb{R}^n} \left\{ \sum_{i=1}^{n} w_i - \frac{1}{\alpha_{\mathrm{J}}} \log\det(A^\top S_x^{-1} W^{\alpha_{\mathrm{J}}} S_x^{-1} A) - \beta_{\mathrm{J}} \sum_{i=1}^{n} \log w_i \right\}, \tag{4.9}$$

with $\beta_{\mathrm{J}} := d/2n$ and $\alpha_{\mathrm{J}} := 1 - 1/\log_2(1/\beta_{\mathrm{J}})$. Lee and Sidford[107] proposed the convex program (4.9) associated with the *approximate John weights* ζ_x, with the aim of searching for the best member of a family of volumetric barrier functions. They analyzed the use of the John weights in the context of speeding up interior point methods for solving linear programs; here we consider them for improving the mixing time of a sampling algorithm. The convex program (4.9) is closely related to the problem of finding the largest ellipsoid at any interior point of the polytope, such that the ellipsoid is contained within the polytope. This problem of finding the largest ellipsoid was first studied by John [92] who showed that each convex body in $\mathbb{R}^d$ contains a unique ellipsoid of maximal volume. The convex program (4.9) was used by Lee and Sidford [107] to compute approximate John Ellipsoids for solving linear programs. In a recent work, Gustafson et al. [74] make use of the exact John ellipsoids and design a polynomial time sampling algorithm for polytopes. See Table 4.1 for the associated guarantees.

Hit-and-run: We conclude with a brief discussion with another popular sampling algorithm: Hit-and-run. It was introduced by Smith [182] as a sampling algorithm for general distributions and it was later shown to have polynomial mixing time for sampling from convex sets [115, 121, 120]. The algorithm proceeds as follows: when at point x, it firsts draws a random line through x and then samples from the one-dimensional marginal of the target distribution restricted to this line. For uniform sampling from convex sets, the second step simplifies to drawing a uniform point from the line restricted to the convex set. Mixing time bounds for this random walk are summarized in Table 4.1.

4.2.2 Mixing time comparisons of walks

Table 4.1 provides a summary of the mixing time bounds and per step complexity and the effective per sample complexity for various random walks, including the Vaidya and John walks analyzed in this chapter. In addition to the Ball Walk, Hit-and-Run, Dikin, Vaidya and John walks, we also show scalings for the recently introduced Riemannian Hamiltonian Monte Carlo (RHMC) on polytopes by [110] and the John's walk based on exact John ellipsoids studied by [74]. The details of per iteration cost for the new random walks is discussed in Section 4.4.1. We now compare and contrast the complexities of these random walks.

Unlike the Ball Walk or hit-and-run which are useful for general convex sets, the Dikin, Vaidya, John and RHMC walks are specialized for polytopes. These latter random walks exploit the definition of the polytope in a particular way so that the transition probability from a point x to y does not change under an affine transformation, i.e., $\mathcal{T}(x, y) = \mathcal{T}(Ax, Ay)$ where $\mathcal{T}$ denotes the transition kernel for the random walk. Consequently, the mixing time bounds for these random walks have no dependence on the condition number of the set κ (4.2). We can see from Table 4.1, that compared to the Ball walk and hit-and-run, Vaidya walk mixes significantly faster if $n \ll d\kappa^2$. The condition number κ of polytopes with polynomially many faces can not be $\mathcal{O}(d^{\frac{1}{2}-\epsilon})$ for any $\epsilon > 0$ but can be arbitrarily larger, even exponential in dimension d [97]. For such polytopes, Vaidya walk mixes faster as long as $n \ll d^3$ (and even for larger n when κ is large). It takes $\mathcal{O}(\sqrt{n/d})$ fewer steps compared to Dikin walk and thus provides a practical speed up over all range of d.

From a warm start, the Riemannian Hamiltonian Monte Carlo on polytopes introduced by [110] has $O\left(nd^{2/3}\right)$ mixing time, and thus mixes faster (up to constants) compared than the Vaidya walk (respectively the John walk) when the number of constraints n is is bounded as $n \ll d^{5/3}$ (respectively $n \ll d^{11/6}$). For larger numbers of constraints, the Vaidya and John walks exhibit faster mixing. More generally, it is clear that the rate of John walk has *almost* the best order across all the walks for reasonably large values of $n \gg d^2$.

Finally, we discuss the (exact) John's walk of [74] with the (approximate) John walk studied in our work. The mixing time of their random walk is remarkably independent of the number of constraints and the per iteration cost also depends linearly on the

number of constraints. Nonetheless, the dependence on d, for both the mixing time (d^7) and the per iteration cost ($nd^4 + d^8$) is quite poor. In contrast, the per iteration cost for our John walk is nd^2 and the mixing time has only a poly-logarithmic dependence on n.

Random walk	$\tau_1(\delta; \mu_0)$	**Iteration cost**	**Per sample cost**
Ball walk$^{\#}$ [94]	$d^2\kappa^2$	nd	$nd^3\kappa^2$
Hit-and-Run [120]	$d^2\kappa^2$	nd	$nd^3\kappa^2$
Dikin walk [97]	nd	nd^2	n^2d^3
RHMC walk [109]	$nd^{2/3}$	nd^2	$n^2d^{2.67}$
John's walk† [74]	d^7	$nd^4 + d^8$	$nd^{11} + d^{15}$
Vaidya walk (this chapter)	$n^{1/2}d^{3/2}$	nd^2	$n^{1.5}d^{3.5}$
John walk (this chapter)	$d^{5/2}\log^4\left(\frac{2n}{d}\right)$	$nd^2\log^2 n$	$nd^{4.5}$
Improved John walk‡ (this chapter)	$d^2\,\kappa_{n,d}$	$nd^2\log^2 n$	nd^4

Table 4.1. Upper bounds on computational complexity of random walks on the polytope $\mathcal{K} = \{x \in \mathbb{R}^d | Ax \leq b\}$ defined by the matrix-vector pair $(A, b) \in \mathbb{R}^{n\times d} \times \mathbb{R}^n$ with a warm-start. For simplicity, here we ignore the logarithmic dependence on the warmness parameter and the tolerance δ. The iteration cost terms of order nd^2 arise from linear system solving, using standard and numerically stable algorithms, for n equations in d dimensions; algorithms with best possible theoretical complexity nd^ω for $\omega < 1.373$ are not numerically stable enough for practical use. $^{\#}$Mixing time of the Ball walk has been improved to $O\left(d^2\kappa\right)$ for near isotropic convex bodies by [111]. While ball walk, Hit-and-run are affected by the condition number κ of the set, the Dikin and RHMC walks have quadratic dependence on the number of constraints n. †John's walk by [74] (based on the exact John ellipsoids) has linear dependence on n but poor dependence on d. In contrast, the Vaidya walk has sub-quadratic dependence on n and significantly better dependence on d. Furthermore, the John walk (based on approximate John's ellipsoids) analyzed in this chapter has linear dependence with reasonable dependence on the dimensions d. ‡The mixing time bound for the improved John walk with poly-logarithmic factor $\kappa_{n,d}$ is conjectured.

4.2.3 Visualization of three walks' proposal distributions

To gain intuition about the three interior point based methods—namely, the Dikin, Vaidya and John walks—it is helpful to discuss how their underlying proposal distributions change as a function of the current point x. All three walks are based on Gaussian

proposal distributions with inverse covariance matrices of the general form

$$\sum_{i=1}^{n} w_{x,i} \frac{a_i a_i^\top}{s_{x,i}^2},$$

where $w_{x,i} > 0$ corresponds to a state-dependent weight associated with the i-th constraint. The Dikin walk uses the weights $w_{x,i} = 1$; the Vaidya walk uses the weights $w_{x,i} = \sigma_{x,i} + \beta_{\mathrm{V}}$; and the John walk uses the weights $w_{x,i} = \zeta_{x,i}$. For simplicity, we refer to these weights as the Dikin, Vaidya and John weights. The i-th weight characterize the importance of the i-th linear constraint in constructing the inverse covariance matrix. A larger value of the weight $w_{x,i}$ relative to the total weight $\sum_{i=1}^{n} w_{x,i}$ signifies more importance for the i-th linear constraint for the point x.

Figure 4.1a illustrates the difference in three weights as we move points inside the polytope $[-1, 1]^2$. When the point x is in the middle of the unit square formed by the four constraints, all walks exhibit equal weight for every constraint. When the point x is closer to the bottom-left boundary, the Vaidya and John weights assign larger weights to the bottom and the left constraints, while the weights for top and right constraints decrease. Note that the total sum of Vaidya weights and that of John weights remains constant independent of the position of the point x.

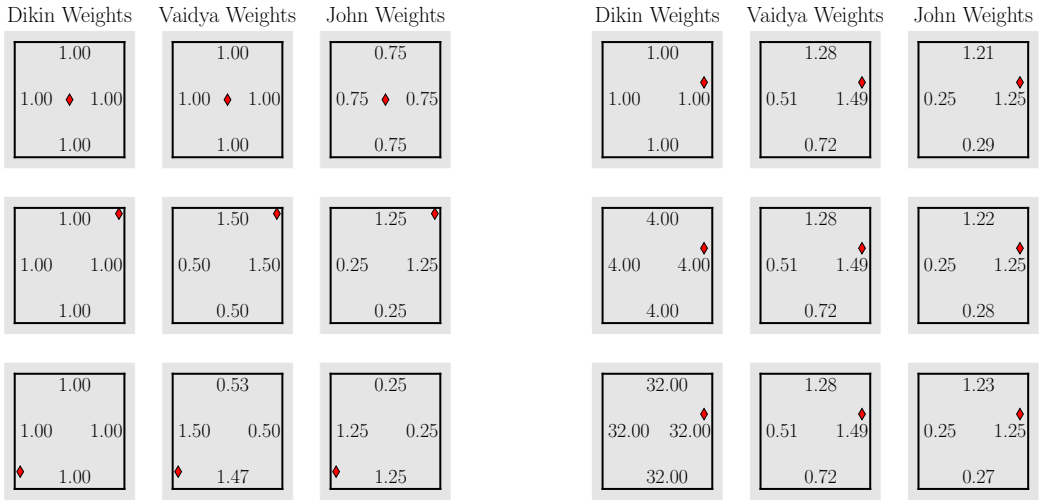
In Figure 4.1b-4.2b, we demonstrate that the Vaidya walk and the John walk are better at handling repeated constraints. Note that we can define the square $[-1, 1]^2$ as

$$[-1, 1]^2 = \left\{ x \in \mathbb{R}^2 \middle| Ax \leq b, A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix}, b = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\}. \tag{4.10}$$

Simply repeating the rows of the matrix A several times changes the mathematical formulatiton of the polytope, but does not change the shape of the polytope. We define the square with constraints repeated $n/4$ times $\mathcal{S}_{n/4}$ as

$$\mathcal{S}_{n/4} = \left\{ x \in \mathbb{R}^2 \middle| A_{n/4} x \leq b_{n/4}, A_{n/4} = \begin{bmatrix} A \\ \vdots \\ \times(n/4) \end{bmatrix}, b_{n/4} = \begin{bmatrix} b \\ \vdots \\ \times(n/4) \end{bmatrix}, \right\} \tag{4.11}$$

where A and b were defined above. We denote effective weight for each distinct constraint as the sum of weights corresponding to the same constraint. Using this definition, the effective Dikin weight, which is $n/4$, is thus affected by the repeating of constraints. Consequently, the Dikin ellipsoid is much smaller for polytopes with repeated constraints. However, the Vaidya and John weights do not change as observed in the Figure 4.1b. Such a property of these two weights implies that the Vaidya and John ellipsoids are not too small even for very large number of constraints. And we observe such a phenomenon in Figures 4.2a-4.2b where the repetition of rows in the matrix A leads to very small Dikin ellipsoid but large Vaidya and John ellipsoid. A



(a) Weights for different locations and a fixed number of constraints n.

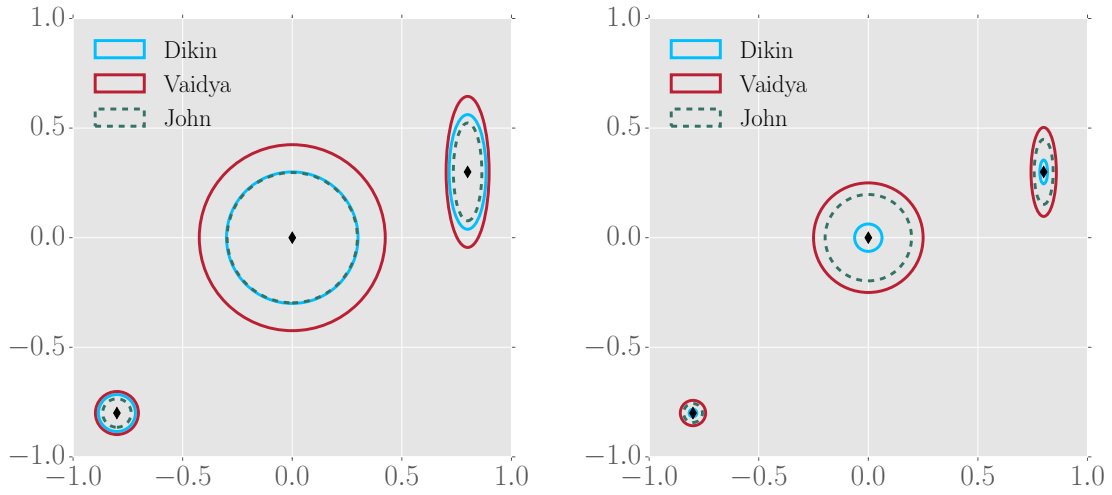
(b) Effective weights for a fixed location and different number of constraints n

Figure 4.1. Visualization of the weights on the square with repeated constraints $\mathcal{S}_{n/4}$ for the different random walks. The number mentioned next to the boundary lines denotes the effective weight for the location x (denoted by diamond) for the corresponding constraint. **(a)** $n = 4$ is common across rows and $x = (0, 0)$ for the top row, $(0.9, 0.9)$ for the middle and $(-0.9, -0.7)$ for the bottom row. The Dikin weights are independent of x, the Vaidya and the John weights for a constraint increase if the location x is closer to it. **(b)** $x = (0.85, 0.30)$ is common across rows, and $n = 4$ for the top row, $n = 16$ for the middle and $n = 128$ for the bottom row. The effective Dikin weight for each constraint increases linearly with n but for the Vaidya and John walk adaptively, the weights get adjusted such that the sum of their weights is always of the order of the dimension d.

few other numerical computations also suggest that the Vaidya and John ellipsoids are moder adaptive when compared to Dikin ellipsoids when the number of constraints is large. Nonetheless, such a claim is only based on heuristics and is presented simply to provide an intuition that the new ellipsoids are better behaved than Dikin ellipsoids and thereby motivated the design of the new random walks.

4.3 Convergence of Vaidya and John walks

With the basic background in place, we now describe the algorithms more precisely and state upper bounds on the mixing time of the Vaidya and John walks. In Section 4.3.4, we propose a variant of the John walk, known as the *improved John walk*, and conjecture that it has a better mixing time bound than that of the John walk.



(a) $n = 32$ (b) $n = 2048$

Figure 4.2. Visualization of the proposal distribution on the square with repeated constraints $\mathcal{S}_{n/4}$ for the different random walks. **(a, b)** Unit ellipsoids associated with the covariances of the random walks at different states x on the square with repeated constraints $\mathcal{S}_{n/4}$. Clearly, all these ellipsoids adapt to the boundary but increasing n has a profound impact on the volume of the Dikin ellipsoids and comparatively less impact on the Vaidya and John ellipsoids.

4.3.1 Vaidya and John walks

In this subsection, we formally define the Vaidya and John walks. In Algorithm 4 and Algorithm 5, we summarize the steps of the Vaidya walk and the John walk.

Vaidya walk: The Vaidya walk with radius parameter $r > 0$, denoted by $\mathrm{VW}(r)$ for short, is defined by a Gaussian proposal distribution denoted as $\mathcal{P}_x^{\mathrm{V}}$: given a current state $x \in \mathrm{int}\,(\mathcal{K})$, it proposes a new point by sampling from the multivariate Gaussian distribution $\mathcal{N}\left(0, \frac{r^2}{\sqrt{nd}} {V_x}^{-1}\right)$. In analytic terms, the proposal density at x is given by

$$\rho_x^{\mathrm{V}}(z) := \rho_{\mathrm{Vaidya}(r)}(x, z) = \sqrt{\det V_x} \left(\frac{nd}{2\pi r^2}\right)^{d/2} \exp\left(-\frac{\sqrt{nd}}{2r^2} (z-x)^\top V_x (z-x)\right). \tag{4.12}$$

As the target distribution for our walk is the uniform distribution on $\mathcal{K}$, the proposal step is followed by an accept-reject step as described in equation 2.7. Thus the overall transition distribution for the walk at state x is defined by a density given by

$$q_{\mathrm{Vaidya}(r)}(x, z) = \begin{cases} \min\left\{\rho_x^{\mathrm{V}}(z), \rho_z^{\mathrm{V}}(x)\right\}, & z \in \mathcal{K} \text{ and } z \neq x, \\ 0, & z \notin \mathcal{K}, \end{cases}$$

and a probability mass at x, given by $1 - \int_{z \in \mathcal{K}} \min\{\rho_x(z), \rho_z(x)\}\, dz$. We use $\mathcal{T}_{\text{Vaidya}(r)}$ to denote the resulting transition operator for the Vaidya walk with parameter r.

Algorithm 4: Vaidya Walk with parameter r (VW(r))

Input: Parameter r and $x_0 \in \text{int}(\mathcal{K})$
Output: Sequence $x_1, x_2, \ldots$

for $i = 0, 1, \ldots$ **do**
  With probability $\frac{1}{2}$ stay at the current state: $x_{i+1} \leftarrow x_i$ % *lazy step*
  With probability $\frac{1}{2}$ perform the following update:
    Proposal step: Draw $z_{i+1} \sim \mathcal{N}\left(x_i, \frac{r^2}{(nd)^{1/2}} V_{x_i}^{-1}\right)$
    Accept-reject step:
      if $z_{i+1} \notin \mathcal{K}$ *then* $x_{i+1} \leftarrow x_i$ % *reject an infeasible proposal*
      else
        compute $\alpha_{i+1} = \min\left\{1, \rho_{z_{i+1}}(x_{i+1}) / \rho_{x_{i+1}}(z_{i+1})\right\}$
        With probability α_{i+1} accept the proposal: $x_{i+1} \leftarrow z_{i+1}$
        With probability $1 - \alpha_{i+1}$ reject the proposal: $x_{i+1} \leftarrow x_i$
end

John walk: The John walk is similar to the Vaidya walk except that the proposals at state $x \in \text{int}(\mathcal{K})$ are generated from the multivariate Gaussian distribution $\mathcal{N}\left(0, \frac{r^2}{d^{3/2} \cdot \log_2^4(2n/d)} {J_x}^{-1}\right)$, where the matrix J_x is defined by equation (4.8), and $r > 0$ is a constant. The proposal distribution at $x \in \text{int}(\mathcal{K})$ is denoted as $\mathcal{P}_x^{\text{J}}$. The proposal step is then followed by an accept-reject step similarly defined as in the Vaidya walk. We use $\mathcal{T}_{\text{John}(r)}$ to denote the resulting transition operator for the John walk with parameter r.

4.3.2 Mixing time bounds for warm start

We are now ready to state an upper bound on the mixing time of the Vaidya walk. In this and other theorem statements, we use c to denote a universal positive constant. Recall that Π^* denotes the uniform distribution on the polytope $\mathcal{K}$, and, that $\mathcal{T}_{\text{Vaidya}(r)}$ denotes the operator on distributions associated with the Vaidya walk.

Theorem 5. *Let μ_0 be any distribution that is ϖ-warm with respect to Π^* as defined in equation (2.12). For any $\delta \in (0, 1]$, the Vaidya walk with parameter $r_V = 10^{-4}$ satisfies*

$$d_{TV}\left(\mathcal{T}^k_{Vaidya(r_V)}(\mu_0), \Pi^*\right) \leq \delta \qquad \text{for all } \; k \geq c n^{1/2} d^{3/2} \log\left(\frac{\sqrt{\varpi}}{\delta}\right). \tag{4.13}$$

The proof of Theorem 5 is provided in Section 4.5. Theorem 5 precisely quantifies the dependence of mixing time of the Vaidya walk on many parameters of interest

Algorithm 5: John Walk with parameter r (JW(r))

Input: Parameter r and $x_0 \in \text{int}(\mathcal{K})$
Output: Sequence $x_1, x_2, \ldots$

for $i = 0, 1, \ldots$ **do**
  With probability $\frac{1}{2}$ stay at the current state: $x_{i+1} \leftarrow x_i$ % *lazy step*
  With probability $\frac{1}{2}$ perform the following update:
    Proposal step: Draw $z_{i+1} \sim \mathcal{N}\left(x_i, \frac{r^2}{d^{3/2}} J_{x_i}^{-1}\right)$ % *this step is different than the Vaidya walk*
    Accept-reject step:
      if $z_{i+1} \notin \mathcal{K}$ *then* $x_{i+1} \leftarrow x_i$ % *reject an infeasible proposal*
      else
        compute $\alpha_{i+1} = \min\left\{1, \rho_{z_{i+1}}(x_{i+1})/\rho_{x_{i+1}}(z_{i+1})\right\}$
        With probability α_{i+1} accept the proposal: $x_{i+1} \leftarrow z_{i+1}$
        With probability $1 - \alpha_{i+1}$ reject the proposal: $x_{i+1} \leftarrow x_i$
end

such as dimension d, number of constraints n, the error tolerance δ and the warmness ϖ. The specific choice $r_{\text{V}} = 10^{-4}$ is for theoretical purposes; in practice, we find that substantially larger values can be used.[2] Our upper bound for the mixing time of the Vaidya walk has $\mathcal{O}(\sqrt{n/d})$ improvement over the current best upper bound for the mixing time of the Dikin walk. In Section 4.4.1, we show that the per iteration cost for the two walks is of the same order. Since $n \geq d$ for closed polytopes in $\mathbb{R}^d$, the effective cost until convergence (iteration complexity multiplied by number of iterations required) for the Vaidya walk is at least of the same order as of the Dikin walk, and significantly smaller when $n \gg d$. Comparing the provable mixing time upper bounds, the Vaidya walk has an advantage over the Dikin walk for the problems where the number of constraints is significantly larger than the number of variables involved. Our simulations also confirm this theoretical finding.

Let us now state our result for the mixing time of the John walk:

Theorem 6. *Suppose that* $n \leq \exp(\sqrt{d})$*, and let* μ_0 *be any distribution that is* ϖ*-warm with respect to* Π^**. Then for any* $\delta \in (0, 1]$*, the John walk with parameter* $r_J = 10^{-5}$ *satisfies*

$$d_{TV}\left(\mathcal{T}^k_{John(r_J)}(\mu_0), \Pi^*\right) \leq \delta \qquad \text{for all } k \geq c\, d^{2.5} \log^4\left(\frac{n}{d}\right) \log\left(\frac{\sqrt{\varpi}}{\delta}\right).$$

[2] A larger than optimal r leads to an undesirable high rejection rate. In practice, we can fine tune r by performing a binary search over the interval $[10^{-4}, 1]$ and keeping track of the rejection rate of the samples during the run of the Markov chain for a given choice of r. A choice of $r > 1$ is obviously bad because then the Vaidya ellipsoid will have poor overlap with polytopes near the boundary, causing high rejection rate and slow down of the chain.

The proof of Theorem 6 is provided in Appendix B.4. Again the specific choice of $r_{\text{J}} = 10^{-5}$ is for theoretical purpose; in practice larger choices are possible. Note that the mixing time bound for the John walk depends only on the number of constraints n via a logarithmic factor, and so is almost independent of n. Consequently, it has a mixing time that is polynomial in d even if the number of constraints n scales exponentially in $\sqrt{d}$. Further, we show in Section 4.4.1 that the cost to execute one step of the John walk is of the same order as of the Dikin walk up to a poly-logarithmic factor in n. Thus, using John walk, we obtain improved mixing time bounds for the case when $n \gg d^2$.

4.3.3 Mixing time bounds from deterministic start

The mixing time bounds in Theorem 5 and 6 depend on the warmness ϖ of the initial distribution. In some applications, it may not be easy to find an ϖ-warm initial distribution. In such cases, we can consider starting the random walk from a deterministic point $x_0 \in \text{int}\,(\mathcal{K})$ that is not too close to the boundary $\partial\mathcal{K}$. Indeed, such a point can be found using standard optimization methods—e.g., using a Phase-I method for Newton's algorithm (see Section 11.5.4 in [21]).

Given such a deterministic initialization, our mixing time guarantees depend on the distance of the starting point from the boundary. This dependence involves the following notion of s-centrality:

Definition 2. *A point $x \in \text{int}\,(\mathcal{K})$ is called* s-central *if for any chord $\overline{ef}$ with end points $e, f \in \partial\mathcal{K}$ passing through x, we have $\|e - x\|_2 / \|f - x\|_2 \leq s$.*

Assuming that it is started at an s-central point x_0, the Dikin walk [97] has a polynomial mixing time. The authors showed that when the walk moves to a new state for the first time, the distribution of the iterate is $O\left((\sqrt{n}s)^d\right)$-warm with respect to the distribution[3] Π^*. Since only constant number of steps is required to get a warm start, for a deterministic start, we can just use the Dikin walk in the beginning to provide a warm start to the Vaidya (or John) walk. This motivates us to define the following hybrid walk.

Given an s-central point x_0, simulate the Dikin walk until we observe a new state. Note that due to *laziness* and the accept-reject step, the chain can stay at the starting point for several steps before making the first move a new state. Let k_1 denote the (random) number of steps taken to make the first move to a new state. After k_1 steps, we run the walk $\text{VW}(r)$ with x_{k_1} as the initial point. We call such a walk as *s-central Dikin-start-Vaidya-walk* with parameter r. Let $\mathcal{T}_{\text{Dikin}}$ denote the transition kernel of the Dikin walk stated above. Then, we have the following mixing time bound for this hybrid walk.

[3]Obtaining a warmness result for the Vaidya walk from a deterministic start from a central point is non-trivial and it is quite possible that the warmness does not improve. As a result, we simply invoke the established result for the Dikin walk.

Corollary 4. *Any s-central Dikin-start-Vaidya-walk with parameter $r = 10^{-4}$ satisfies*

$$d_{TV}\left(\mathcal{T}^{k}_{Vaidya(r)}\left(\mathcal{T}^{k_1}_{Dikin}(\delta_{x_0})\right), \Pi^*\right) \leq \delta \qquad \text{for all } k \geq cn^{1/2}d^{5/2}\log\left(\frac{ns}{\delta}\right),$$

where k_1 is a geometric random variable with $\mathbb{E}[k_1] \leq c'$, and $c, c' > 0$ are universal constants.

The mixing rate is logarithmic in ns and has an extra factor of d compared to the bounds in Theorem 5. However, guaranteeing a warm start for a general polytope is hard but obtaining a central point involves only a few steps of optimization. Consequently, the hybrid walk and the guarantees from Corollary 4 come in handy for all such cases. Once again we observe that the upper bounds for mixing time are improved by a factor of $\mathcal{O}(\sqrt{n/d})$ when compared to the Dikin walk from an s-central start [97, 141] which had a mixing time of $O(nd^2)$. The proof follows immediately from Theorem 1 by Kannan and Narayanan [97] and Theorem 5 of this chapter and is thereby omitted.

In a similar fashion, we can provide a polynomial time guarantee for a modified John walk from a deterministic start. We can consider a hybrid random walk that starts at an s-central point, simulates the Dikin walk until it makes the first move to a new state, and from there onwards simulates the John walk. Such a chain would have a mixing time of $O(d^{3.5}\text{poly-log}(n, d, s))$. For brevity, we omit a formal statement of this result.

4.3.4 Conjecture on improved John walk

From our analysis, we suspect that it is possible to improve the mixing time bound of $O(d^{2.5}\text{poly-log}(n/d))$ in Theorem 6 by considering a variant of the John walk. In particular, we conjecture that a random walk with proposal distribution given by $\mathcal{N}\left(x, \frac{r^2}{d\cdot\text{poly-log}(n/d)}{J_x}^{-1}\right)$ for a suitable choice of r has an $O(d^2\text{poly-log}(n/d))$ mixing time from a warm start. We refer to this random walk as the *improved John walk*, and denote its transition operator by $\mathcal{T}_{\text{John}^+}$. Let us now give a formal statement of our conjecture on its mixing rate.

Conjecture 1. *Let μ_0 be any ϖ-warm distribution. Then for any $\delta \in (0, 1]$, the improved John walk with parameter $r = r_0$, satisfies the bound*

$$d_{TV}\left(\mathcal{T}^{k}_{John^+}(\mu_0), \Pi^*\right) \leq \delta \qquad \text{for all } k \geq c\, d^2\, \log_2^{c'}\left(\frac{2n}{d}\right)\log\left(\frac{\sqrt{\varpi}}{\delta}\right),$$

where r_0, c, c' are universal constants.

Note that this conjecture involves quadratic (degree two) scaling in d; this exponent of two matches the sum of exponents for d and n in the mixing time bounds for both the Dikin and Vaidya walks from a warm-start. Consquently, the improved John walk would have better performance than the Dikin, Vaidya and John walks for almost all ranges of (n, d), apart from possible poly-logarithmic factors in the ratio n/d.

4.3.5 Proof sketch

In this subsection, we provide a high-level sketch of the main ingredients of the main proof. It is well-known that mixing of a Markov chain is closely related to its *conductance*. Our main proof relies on the work by [115] that characterizes the conductance of Markov chains on a convex set using Hilbert metric. Precisely, [115] showed that a Markov chain has good conductance if it makes jumps to regions with large overlaps from two nearby points and the mixing time depends inversely on the maximum Hilbert metric between such nearby points. Using this argument, it remains to make sure that the ellipsoid radius is chosen properly such that the ellipsoids remain inside the polytope and the ellipsoids corresponding to two different points x and y overlap a lot even if the points x and y are relatively far apart.

The conductance-based argument has been used for analyzing the ball walk [119, 118], Hit-and-run [115, 120] and the Dikin walk [97, 141, 173]. We refer the reader to the survey by Vempala [193] for a thorough discussion about the relation between the conductance and mixing time for Markov chains. Our proof techniques share a few features with the recent analyses of the Dikin walk by Kannan and Narayanan [97] and [173]. However, new technical ideas are needed in order to handle the state-dependent weights σ_x (4.6b) and ζ_x (4.9) that underlie the proposal distributions for the Vaidya and John walks. Note that these techniques are not present in the analysis of the Dikin walk, which is based on constant weights.

Specifically, we present the proof of Theorem 5 on the mixing time of the Vaidya walk in Section 4.5 and defer the intermediate technical results to Appendix B.1, B.2 and B.3. We present the proof of Theorem 6 (mixing time bound for the John walk) in Appendix B.4 and provide related auxiliary results and their proofs in Appendices B.5, B.6, B.7, B.8 and B.9. As alluded to earlier, to keep the thesis self-contained we provide the proof of Lovász's Lemma in Appendix B.10.

4.4 Numerical experiments

In this section, we first analyze the per-iteration cost to implement of three walks. We show that while the Dikin walk has the best per-iteration cost, the per-iteration cost of the Vaidya walk is only twice of that of Dikin walk and the per-iteration cost of the John walk is only of order $\log_2(2n/d)$ larger. Second, we demonstrate the speed-up gained by the Vaidya walk over the Dikin walk for a warm start on different polytopes.

4.4.1 Per iteration cost

We now show that the per iteration cost of the Dikin, Vaidya and John walks is of the same order. The proposal step of Vaidya walk requires matrix operations like matrix inversion, matrix multiplication and singular value decomposition (SVD). The accept-reject step requires computation of matrix determinants, besides a few matrix

inverses and matrix-vector products. The complexity of all aforementioned operations is $O(nd^2)$. Thus, per iteration computational complexity for the Vaidya walk is $O(nd^2)$.[4]

Both the Dikin and Vaidya walks requires an SVD computation for inverting the Hessian of Dikin barrier $\nabla^2\mathcal{F}_x$. In addition for the Vaidya walk, we have to invert the matrix V_x, which leads to almost twice the computation time of the Dikin walk per step. This difference can be observed in practice.

For the John walk we need to compute the weights ζ_x at each point which involves solving the program (4.9). [107] argued that the convex program (4.9) for obtaining John walk's weights is strongly convex under appropriate norm. They proved that solving this program requires $\log^2 n$ number of gradient steps where each gradient step has the computational complexity of a linear system solve ($O(nd^2)$ using a numerically stable routine). Thus, the overall cost for the John walk is of the same order as of the Dikin walk up to a poly-logarithmic factor in the pair (n, d).

In practice, for the John walk, the combined effect of logarithmic factors in the number of steps and the cost to implement each step is pretty significant. This extra factor becomes a bottleneck for the overall run time for the convergence of the Markov chain. Consequently, the John walk is not suitable for polytopes with moderate values of n and d, and its mixing time bounds are computationally superior to the Dikin and Vaidya walks only for the polytopes with $n \gg d \gg 1$.

4.4.2 Simulations

We now present simulation results for the random walks in $\mathbb{R}^d$ for $d = 2, 10$ and 50 with initial distribution $\mu_0 = \mathcal{N}(0, \sigma_d^2 \mathbb{I}_d)$ and target distribution being uniform, on the following polytopes:

Set-up 1 : The set $[-1, 1]^2$ defined by different number of constraints.

Set-up 2 : The set $[-1, 1]^d$ for $d \in \{2, 3, 4, 5, 6, 7\}$ for $n = \{2d, 2d^2, 2d^3\}$ constraints.

Set-up 3 : Symmetric polytopes in $\mathbb{R}^2$ with n-randomly-generated-constraints.

Set-up 4 : The interior of regular n-polygons on the unit circle.

Set-up 5 : Hyper cube $[-1, 1]^d$ for $d = 10$ and 50.

We choose σ_d such that the warmness parameter M is bounded by 100. We provide implementations of the Dikin, Vaidya and John walks in python and a jupyter notebook at the github repository https://github.com/yuachen/polytopewalk.

We use the following three ways to compare the convergence rate of the Dikin and the Vaidya walks: (1) comparing the approximate mixing time of a particular subset of the polytope—smaller value is associated with a faster mixing chain; (2) comparing the plot of the empirical distribution of samples from multiple runs of the Markov chain

[4] In theory, the matrix computations for the Dikin walk can be carried out in time nd^ν for an exponent $\nu < 1.373$, but such algorithms are not numerically stable enough for practical use.

after k steps—if it appears *more uniform* for smaller k, the chain is deemed to be faster; and (3) contrasting the sequential plots of one dimensional projection of samples for a single long run of the chain—*less smooth* plot is associated with effective and fast exploration leading to a faster mixing [207]. Note that MCMC convergence diagnostics is a hard problem, especially in high dimensions, and since the methods outlined above are heuristic in nature we expect our experiments to not fully match our theoretical results.

In **Set-up 1**, we consider the polytope $[-1, 1]^2$ which can be represented by exactly 4 linear constraints (see Section 4.2.3). Suppose that we repeat the rows of the matrix A, and then run the Dikin and Vaidya walks with the new A. Given the larger number of constraints, our theory predicts that the random walks should mix more slowly. In Figure 4.3c and 4.3d, we plot the empirical distribution obtained by the Dikin walk and Vaidya walk, starting from 200 i.i.d initial samples, for $n = 64$ and 2048. The empirical distribution plot shows that having large n significantly slows the mixing rate of the Dikin walk, while the effect on the Vaidya walk is much less. Further, we also plot the scaling of the approximate mixing time $\hat{k}_{\text{mix}}$ (defined below) for this simulation as a function of the number of constraints n in Figure 4.3b. For **Set-up 2**, we plot $\hat{k}_{\text{mix}}$ as a function of the dimensions d in Figures 4.3e-4.3g, for the random walks on $[-1, 1]^d$ where the hypercube is parametrized by different number of constraints $n \in \{2d, 2d^2, 2d^3\}$. The approximate mixing time is defined with respect to the set $S_d = \{x \in \mathbb{R}^d | \, |x_i| \geq c_d \; \forall i \in [d]\}$ where c_d is chosen such that $\Pi^*(S_d) = 1/2$. In particular, for a fixed value of n, let $\hat{\mathcal{T}}^k$ denote the empirical measure after k-iterations across 2000 experiments. The approximate mixing time $\hat{k}_{\text{mix}}$ is defined as

$$\hat{k}_{\text{mix}} := \min \left\{ k \middle| \Pi^*(S_d) - \hat{\mathcal{T}}^k(S_d) \leq \frac{1}{20} \right\}, \tag{4.14}$$

We choose such a set since the set covers the regions near to the boundary of the polytope which are not covered well by the chosen initial distribution. We make the following observations:

1. The slopes of the best-fit lines, for $\hat{k}_{\text{mix}}$ versus n in the log-log plot in Figure 4.3b, are 0.88 and 0.45 for Dikin and Vaidya walks respectively. This observation reflects a near-linear and sub-linear dependence on n for a fixed d for the mixing time of the Dikin walk and the Vaidya walk respectively.

2. In Figures 4.3e-4.3g, once again we observe a more significant effect of increasing the number of constraints on the approximate mixing time $\hat{k}_{\text{mix}}$. We list the slopes of the best fit lines on these log-log plots in Table 4.2. These slopes correspond to the exponents for d for the approximate mixing time. From the table, we can observe that these experiments agree with the mixing time bounds of $O(nd)$ for the Dikin walk and $O(n^{0.5}d^{1.5})$ for the Vaidya walk.

In **Set-up 3**, we compare the plots of the empirical distribution of 200 runs of the Dikin walk and the Vaidya walk for different values of k, for symmetric polytopes in

No. of Constraints	DW Theoretical	VW Theoretical	DW Experiments	VW Experiments
$n = 2d$	2.0	2.0	1.58	1.72
$n = 2d^2$	3.0	2.5	2.80	2.48
$n = 2d^3$	4.0	3.0	3.84	2.75

Table 4.2. Value of the exponent of dimensions d for the theoretical bounds on mixing time and the observed approximate mixing time of the Dikin walk (DW) and the Vaidya walk (VW) for $[-1, 1]^d$ described by $n = 2d, 2d^2, 2d^3$ constraints. The theoretical exponents are based on the mixing time bounds of $O(nd)$ for the Dikin walk and $O\left(n^{0.5}d^{1.5}\right)$ for the Vaidya walk. The experimental exponents are based on the results from the simulations described in **Set-up 2** in Section 4.4.2. Clearly, the exponents observed in practice are in agreement with the theoretical rates and imply the faster convergence of the Vaidya walk compared to the Dikin walk for large number of constraints.

$\mathbb{R}^2$ with n-randomly-generated-constraints. We fix $b_i = 1$. To generate a_i, first we draw two uniform random variables from $[0, 1]$ and then flip the sign of both of them with probability $1/2$ and assign these values to the vector a_i. The resulting polytope is always a subset of the square $\mathcal{K} = [-1, 1]^2$ and contains the diagonal line connecting the points $(-1, 1)$ and $(1, -1)$. From Figure 4.4a-4.4b, we observe that while there is no clear winner for the case $n = 64$, the Vaidya walk mixes mixes significantly faster than the Dikin walk for the polytope defined by 2048 constraints.

In **Set-up 4**, the constraint set is the regular n-polygons inscribed in the unit circle. A similar observation as in **Set-up 3** can be made from Figure 4.4c-4.4d: the Vaidya walk mixes at least as fast as the Dikin walk and mixes significantly faster for large n.

In **Set-up 5**, we examine the performance of the Dikin walk and the Vaidya walk on hyper-cube $[-1, 1]^d$ for $d = 10, 50$. We plot the one dimensional projections onto a random normal direction of all the samples from a single run up to $10,000$ steps. The Vaidya sequential plot looks more jagged than that of the Dikin walk for $d = 10, n = 5120$. For other cases, we do not have a clear winner. Such an observation is consistent with the $O(\sqrt{n/d})$ speed up of the Vaidya walk which is apparent when the ratio n/d is large.

4.5 Proofs

We begin with auxiliary results in Section 4.5.1 which we use then to prove Theorem 5 in Section 4.5.2. Proofs of the auxiliary results are in Sections 4.5.3 and 4.5.4, and we defer other technical results to appendices.

4.5.1 Auxiliary results

Our proof proceeds by formally establishing the following property for the Vaidya walk: if two points are close, then their one-step transition distribution are also close. Consequently, we need to quantify the closeness between two points and the associated

transition distributions. We measure the distance between two points in terms of the cross ratio that we define next. For a given pair of points $x, y \in \mathcal{K}$, let $e(x), e(y) \in \partial\mathcal{K}$ denote the intersection of the chord joining x and y with $\mathcal{K}$ such that $e(x), x, y, e(y)$ are in order (see Figure 4.6a). The cross-ratio $d_{\mathcal{K}}(x, y)$ is given by

$$d_{\mathcal{K}}(x, y) := \frac{\|e(x) - e(y)\|_2 \, \|x - y\|_2}{\|e(x) - x\|_2 \, \|e(y) - y\|_2}. \tag{4.15}$$

The ratio $d_{\mathcal{K}}(x, y)$ is related to the Hilbert metric on $\mathcal{K}$, which is given by $\log(1 + d_{\mathcal{K}}(x, y))$; see the paper by [27] for more details.

Consider a lazy reversible random walk on a bounded convex set $\mathcal{K}$ with transition operator $\mathcal{T}$ defined via the mapping $\mu_0 \mapsto \mu_0/2 + \tilde{\mathcal{T}}(\mu_0)/2$ and stationary with respect to the uniform distribution on $\mathcal{K}$ (denoted by Π^*). (Recall that δ_x denote the dirac-delta distribution with unit mass at x.) The following lemma gives a bound on the mixing-time of the Markov chain.

Lovász's Lemma. *Suppose that there exist scalars $\varrho, \Delta \in (0, 1)$ such that*

$$d_{TV}\left(\tilde{\mathcal{T}}(\delta_x), \tilde{\mathcal{T}}(\delta_y)\right) \leq 1 - \varrho \qquad \text{for all } x, y \in \operatorname{int}(\mathcal{K}) \text{ with } d_{\mathcal{K}}(x, y) < \Delta. \tag{4.16a}$$

Then for every distribution μ_0 that is M-warm with respect to Π^*, the lazy transition operator $\mathcal{T}$ satisfies

$$d_{TV}\left(\mathcal{T}^k(\mu_0), \Pi^*\right) \leq \sqrt{M} \exp\left(-k \, \frac{\Delta^2 \varrho^2}{4096}\right) \quad \forall \;\; k = 1, 2, \ldots. \tag{4.16b}$$

This result is implicit in the paper by [115], though not explicitly stated. In order to keep the thesis self-contained, we provide a proof of this result in Appendix B.10.

Our proof of Theorem 5 is based on applying Lovász's Lemma; the main challenge in our work is to establish that our random walks satisfy the condition (4.16a) with suitable choices of Δ and ϱ. In order to proceed with the proof, we require a few additional notations. Recall that the slackness at x was defined as $s_x := (b_1 - a_1^\top x, \ldots, b_n - a_n^\top x)^\top$. For all $x \in \operatorname{int}(\mathcal{K})$, define the *Vaidya local norm of v at x* as

$$\|v\|_{V_x} := \left\|V_x^{1/2} v\right\|_2 = \sqrt{\sum_{i=1}^{n} (\sigma_{x,i} + \beta_{\mathrm{V}}) \frac{(a_i^\top v)^2}{s_{x,i}^2}}, \tag{4.17a}$$

and the *Vaidya slack sensitivity at x* as

$$\theta_{V_x} := \left(\left\| \frac{a_1}{s_{x,1}} \right\|_{V_x}^2, \ldots, \left\| \frac{a_n}{s_{x,n}} \right\|_{V_x}^2 \right)^\top = \left(\frac{a_1^\top V_x^{-1} a_1}{s_{x,1}^2}, \ldots, \frac{a_n^\top V_x^{-1} a_n}{s_{x,n}^2} \right)^\top. \tag{4.17b}$$

Similarly, we define the *John local norm of v at x* and the *John slack sensitivity at x* as

$$\|v\|_{J_x} := \left\|J_x^{1/2} v\right\|_2 \quad \text{and} \quad \theta_{J_x} := \left(\left\| \frac{a_1}{s_{x,1}} \right\|_{J_x}^2, \ldots, \left\| \frac{a_n}{s_{x,n}} \right\|_{J_x}^2 \right)^\top. \tag{4.17c}$$

The following lemma provides useful properties of the leverage scores σ_x from equation (4.6b), the weights ζ_x obtained from solving the program (4.9), and the slack sensitivities θ_{V_x} and θ_{J_x}.

Lemma 11. *For any $x \in \text{int}\,(\mathcal{K})$, the following properties hold:*

(a) $\sigma_{x,i} \in [0,1]$ *for all* $i \in [n]$,

(b) $\sum_{i=1}^{n} \sigma_{x,i} = d$,

(c) $\theta_{V_x,i} \in \left[0, \sqrt{n/d}\right]$ *for all* $i \in [n]$,

(d) $\zeta_{x,i} \in [\beta_J, 1+\beta_J]$ *for all* $i \in [n]$,

(e) $\sum_{i=1}^{n} \zeta_{x,i} = 3d/2$, *and*

(f) $\theta_{J_x,i} \in [0,4]$ *for all* $i \in [n]$.

We prove this lemma in Section 4.5.3.

Let $\mathcal{P}_x^{\text{V}}$ to denote the proposal distribution of the random walk VW(r) at state x. Next, we state a lemma that shows that if two points $x, y \in \text{int}\,(\mathcal{K})$ are close in Vaidya local norm at x, then for a suitable choice of the parameter r, the proposal distributions $\mathcal{P}_x^{\text{V}}$ and $\mathcal{P}_y^{\text{V}}$ are close. In addition, we show that the proposals are accepted with high probability at any point $x \in \text{int}\,(\mathcal{K})$. To establish the latter result, we now define the non-lazy transition operator of the Vaidya walk. Since the Vaidya walk is lazy with probability 1/2, there exists a valid (non-lazy) transition operator $\tilde{\mathcal{T}}_{\text{Vaidya}(r)}$ such that for any distribution μ_0, we have

$$\mathcal{T}_{\text{Vaidya}(r)}(\mu_0) = \mu_0/2 + \tilde{\mathcal{T}}_{\text{Vaidya}(r)}(\mu_0)/2.$$

We call $\tilde{\mathcal{T}}_{\text{Vaidya}}$ the non-lazy transition operator for the Vaidya walk. Note that the one-step non-lazy transition distribution $\tilde{\mathcal{T}}_{\text{Vaidya}(r)}(\delta_x)$ denotes the distribution of proposals after the accept-reject step if the chain was not lazy. Thus to establish that proposals are accepted with high probability, it suffices to establish that the transition distribution $\tilde{\mathcal{T}}_{\text{Vaidya}(r)}(\delta_x)$ at any point $x \in \mathcal{K}$ is close to the proposal distribution $\mathcal{P}_x^{\text{V}}$. We now state these two results formally:

Lemma 12. *There exists a continuous non-decreasing function $f : [0, 1/4] \to \mathbb{R}_+$ with $f(1/15) \geq 10^{-4}$ such that for any $\epsilon \in (0, 1/15]$, the random walk VW(r) with $r \in [0, f(\epsilon)]$ satisfies*

$$d_{TV}\left(\mathcal{P}_x^V, \mathcal{P}_y^V\right) \leq \epsilon \quad \forall\, x, y \in \text{int}\,(\mathcal{K}) \text{ s.t. } \|x-y\|_{V_x} \leq \frac{\epsilon r}{2(nd)^{1/4}}\,, \quad \text{and} \tag{4.18a}$$

$$d_{TV}\left(\tilde{\mathcal{T}}_{Vaidya(r)}(\delta_x), \mathcal{P}_x^V\right) \leq 5\epsilon \;\; \forall\; x \in \text{int}\,(\mathcal{K}). \tag{4.18b}$$

See Section 4.5.4 for the proof of this lemma.

With these lemmas in hand, we are now equipped to prove Theorem 5. To simplify notation, for the rest of this section, we adopt the shorthands $\mathcal{T}_x = \tilde{\mathcal{T}}_{\text{Vaidya}(r)}(\delta_x)$, $\mathcal{P}_x = \mathcal{P}_x^{\text{V}}$ and $\|\cdot\|_{V_x} = \|\cdot\|_x$.

4.5.2 Proof of Theorem 5

In order to invoke Lovász's Lemma for the random walk $\text{VW}(10^{-4})$, we need to verify the condition (4.16a) for suitable choices of ϱ and Δ. Doing so involves two main steps:

(A): First, we relate the cross-ratio $d_{\mathcal{K}}(x, y)$ to the local norm (4.17a) at x.

(B): Second, we use Lemma 12 to show that if $x, y \in \text{int}\,(\mathcal{K})$ are close in local-norm, then the transition distributions $\mathcal{T}_x$ and $\mathcal{T}_y$ are close in TV-distance.

Step (A): We claim that for all $x, y \in \text{int}\,(\mathcal{K})$, the cross-ratio can be lower bounded as

$$d_{\mathcal{K}}(x, y) \geq \frac{1}{\sqrt{2d}} \|x - y\|_x . \tag{4.19}$$

Note that we have

$$\begin{aligned} d_{\mathcal{K}}(x, y) = \frac{\|e(x) - e(y)\|_2 \|x - y\|_2}{\|e(x) - x\|_2 \|e(y) - y\|_2} &\overset{(i)}{\geq} \max\left\{ \frac{\|x - y\|_2}{\|e(x) - x\|_2}, \frac{\|x - y\|_2}{\|e(y) - y\|_2} \right\} \\ &\overset{(ii)}{\geq} \max\left\{ \frac{\|x - y\|_2}{\|e(x) - x\|_2}, \frac{\|x - y\|_2}{\|e(y) - x\|_2} \right\}, \end{aligned}$$

where step (i) follows from the inequality $\|e(x) - e(y)\|_2 \geq \max\{\|e(y) - y\|_2, \|e(x) - x\|_2\}$; and step (ii) follows from the inequality $\|e(x) - x\|_2 \leq \|e(y) - x\|_2$. Furthermore, from Figure 4.6b, we observe that

$$\max\left\{ \frac{\|x - y\|_2}{\|e(x) - x\|_2}, \frac{\|x - y\|_2}{\|e(y) - x\|_2} \right\} = \max_{i \in [n]} \left| \frac{a_i^\top (x - y)}{s_{x,i}} \right| . \tag{4.20}$$

This argument of equation (4.11) has also been used [173, lemma 9]. Note that maximum of a set of non-negative numbers is greater than the mean of the numbers. Combining this fact with properties (a) and (b) from Lemma 11, we find that

$$d_{\mathcal{K}}(x, y) \geq \sqrt{\frac{1}{\sum_{i=1}^n (\sigma_{x,i} + \beta_{\text{V}})} \sum_{i=1}^n (\sigma_{x,i} + \beta_{\text{V}}) \frac{(a_i^\top (x - y))^2}{s_{x,i}^2}} = \frac{\|x - y\|_x}{\sqrt{2d}},$$

thereby proving the claim (4.19).

Step (B): By the triangle inequality, we have

$$d_{\mathrm{TV}}\left(\mathcal{T}_x, \mathcal{T}_y\right) \leq d_{\mathrm{TV}}\left(\mathcal{T}_x, \mathcal{P}_x\right) + d_{\mathrm{TV}}\left(\mathcal{P}_x, \mathcal{P}_y\right) + d_{\mathrm{TV}}\left(\mathcal{P}_y, \mathcal{T}_y\right).$$

Thus, for any (r, ϵ) such that $\epsilon \in [0, 1/15]$ and $r \leq f(\epsilon)$, Lemma 12 implies that

$$d_{\mathrm{TV}}\left(\mathcal{T}_x, \mathcal{T}_y\right) \leq 11\epsilon, \quad \forall x, y \in \mathrm{int}\left(\mathcal{K}\right) \text{ such that } \|x - y\|_x \leq \frac{r\epsilon}{2(nd)^{1/4}}.$$

Consequently, the walk $\mathrm{VW}(r)$ satisfies the assumptions of Lovász's Lemma with

$$\Delta := \frac{1}{\sqrt{2d}} \cdot \frac{r\epsilon}{2(nd)^{1/4}} \quad \text{and} \quad \varrho := 1 - 11\epsilon.$$

Since $f(1/15) \geq 10^{-4}$, we can set $\epsilon = 1/15$ and $r = 10^{-4}$, whence

$$\Delta^2 \varrho^2 = \frac{(1 - 11\epsilon)^2 \epsilon^2 r^2}{8d\sqrt{nd}} = \frac{4^2}{15^2} \frac{1}{15^2} \frac{1}{10^{-8}} \cdot \frac{1}{d\sqrt{nd}} \geq 10^{-12} \frac{1}{d\sqrt{nd}}.$$

Observing that $\Delta < 1$ yields the claimed upper bound for the mixing time of Vaidya Walk.

4.5.3 Proof of Lemma 11

In order to prove part (a), observe that for any $x \in \mathrm{int}\left(\mathcal{K}\right)$, the Hessian $\nabla^2 \mathcal{F}_x := \sum_{i=1}^n a_i a_i^\top / s_{x,i}^2$ is a sum of rank one positive semidefinite (PSD) matrices. Also, we can write $\nabla^2 \mathcal{F}_x = A_x^\top A_x$ where

$$A_x := \begin{bmatrix} a_1^\top / s_{x,1} \\ \vdots \\ a_n^\top / s_{x,n} \end{bmatrix}.$$

Since $\mathrm{rank}(A_x) = d$, we conclude that the matrix $\nabla^2 \mathcal{F}_x$ is invertible and thus, both the matrices $\nabla^2 \mathcal{F}_x$ and $\left(\nabla^2 \mathcal{F}_x\right)^{-1}$ are PSD. Since $\sigma_{x,i} = a_i^\top \left(\nabla^2 \mathcal{F}_x\right)^{-1} a_i / s_{x,i}^2$, we have $\sigma_{x,i} \geq 0$. Further, the fact that $a_i a_i^\top / s_{x,i}^2 \preceq \nabla^2 \mathcal{F}_x$ implies that $\sigma_{x,i} \leq 1$.

Turning to the proof of part (b), from the equality $\mathrm{trace}(AB) = \mathrm{trace}(BA)$, we obtain

$$\sum_{i=1}^n \sigma_{x,i} = \mathrm{trace}\left(\sum_{i=1}^n \frac{a_i^\top \left(\nabla^2 \mathcal{F}_x\right)^{-1} a_i}{s_{x,i}^2}\right) = \mathrm{trace}\left(\left(\nabla^2 \mathcal{F}_x\right)^{-1} \sum_{i=1}^n \frac{a_i a_i^\top}{s_{x,i}^2}\right) = \mathrm{trace}(\mathbb{I}_d) = d.$$

Now we prove part (c). Using the fact that $\sigma_{x,i} \geq 0$, and an argument similar to part (a) we find that that the matrices V_x and V_x^{-1} are PSD. Since $\theta_{V_x,i} = a_i^\top V_x^{-1} a_i / s_{x,i}^2$, we have $\theta_{V_x,i} \geq 0$. It is straightforward to see that $\beta_{\mathrm{V}} \nabla^2 \mathcal{F}_x \preceq V_x$ which implies that $\theta_{V_x,i} \leq \sigma_{x,i}/\beta$. Further, we also have $\left(\sigma_{x,i} + \beta_{\mathrm{V}}\right) \frac{a_i a_i^\top}{s_{x,i}^2} \preceq V_x$ and whence $\theta_{V_x,i} \leq 1/\left(\sigma_{x,i} + \beta_{\mathrm{V}}\right)$. Combining the two inequalities yields the claim.

The other parts of the Lemma follow from Lemma 13, 14 and 15 by [107] and are thereby omitted here.

4.5.4 Proof of Lemma 12

We prove the lemma for the following function

$$f(\epsilon) := \min\left\{\frac{1}{20\left(1+\sqrt{2}\log^{\frac{1}{2}}\left(\frac{4}{\epsilon}\right)\right)}, \frac{\epsilon}{\sqrt{18\log(2/\epsilon)}}, \sqrt{\frac{\epsilon}{86\sqrt{3}\chi_2}}, \frac{\epsilon}{22\sqrt{5/3}\chi_3}, \sqrt{\frac{\epsilon}{50\sqrt{105}\chi_4}}\right\}, \tag{4.21}$$

where $\chi_k = (2e/k \cdot \log(4/\epsilon))^{k/2}$ for $k = 2, 3$ and 4. A numerical calculation shows that $f(1/15) \geq 10^{-4}$.

Proof of claim (4.18a)

In order to bound the total variation distance $d_{\mathrm{TV}}(\mathcal{P}_x, \mathcal{P}_y)$, we apply Pinsker's inequality, which provides an upper bound on the TV-distance in terms of the KL divergence:

$$d_{\mathrm{TV}}(\mathcal{P}_x, \mathcal{P}_y) \leq \sqrt{2\mathrm{KL}(\mathcal{P}_x \parallel \mathcal{P}_y)}.$$

For Gaussian distributions, the KL divergence has a closed form expression. In particular, for two normal-distributions $\mathcal{G}_1 = \mathcal{N}(\mu_1, \Sigma_1)$ and $\mathcal{G}_2 = \mathcal{N}(\mu_2, \Sigma_2)$, the Kullback-Leibler divergence between the two is given by

$$\begin{aligned}&\mathrm{KL}(\mathcal{G}_1 \parallel \mathcal{G}_2)\\ &= \frac{1}{2}\Big(\mathrm{trace}(\Sigma_1^{-1/2}\Sigma_2\Sigma_1^{-1/2}) - d - \log\det(\Sigma_1^{-1/2}\Sigma_2\Sigma_1^{-1/2}) + (\mu_1-\mu_2)^\top \Sigma_1^{-1}(\mu_1-\mu_2)\Big).\end{aligned}$$

Recall from equation (4.12) that the proposal distribution for Vaidya walk is Gaussian, i.e., $\mathcal{P}_x = \mathcal{N}\left(x, \frac{r}{\sqrt{nd}}V_x^{-1}\right)$. Substituting $\mathcal{G}_1 = \mathcal{P}_x$ and $\mathcal{G}_2 = \mathcal{P}_y$ into the above expression and applying Pinsker's inequality, we find that

$$\begin{aligned}d_{\mathrm{TV}}(\mathcal{P}_x, \mathcal{P}_y)^2 &\leq 2\mathrm{KL}(\mathcal{P}_y \parallel \mathcal{P}_x)\\ &= \mathrm{trace}(V_x^{-1/2}V_yV_x^{-1/2}) - d - \log\det(V_x^{-1/2}V_yV_x^{-1/2}) + \frac{\sqrt{nd}}{r^2}\|x-y\|_x^2\\ &= \left\{\sum_{i=1}^{d}\left(\lambda_i - 1 + \log\frac{1}{\lambda_i}\right)\right\} + \frac{\sqrt{nd}}{r^2}\|x-y\|_x^2,\end{aligned} \tag{4.22}$$

where $\lambda_1, \ldots, \lambda_d > 0$ denote the eigenvalues of the matrix $V_x^{-1/2}V_yV_x^{-1/2}$, and we have used the facts that $\det(V_x^{-1/2}V_yV_x^{-1/2}) = \prod_{i=1}^d \lambda_i$ and $\mathrm{trace}(V_x^{-1/2}V_yV_x^{-1/2}) = \sum_{i=1}^d \lambda_i$. The following lemma is useful in bounding expression (4.22).

Lemma 13. *For any scalar $t \in [0, 1/12]$ and any pair $x, y \in \mathrm{int}(\mathcal{K})$ under the condition that $\|x-y\|_x \leq t/(nd)^{1/4}$, we have*

$$\left(1 - \frac{8t}{\sqrt{d}}\right)\mathbb{I}_d \preceq V_x^{-1/2}V_yV_x^{-1/2} \preceq \left(1 + \frac{8t}{\sqrt{d}}\right)\mathbb{I}_d,$$

where $\preceq$ denotes ordering in the PSD cone, and $\mathbb{I}_d$ is the d-dimensional identity matrix.

See Appendix B.2 for the proof of this lemma.

For $\epsilon \in (0, 1/15]$ and $r \in [0, 1/12]$, we have $t = \epsilon r/2 \leq 1/12$, whence the eigenvalues $\{\lambda_i, i \in [d]\}$ can be sandwiched as

$$\frac{1}{2} \leq 1 - \frac{4\epsilon r}{\sqrt{d}} \leq \lambda_i \leq 1 + \frac{4\epsilon r}{\sqrt{d}} \quad \text{for all } i \in d. \tag{4.23}$$

We are now ready to bound the TV distance between $\mathcal{P}_x$ and $\mathcal{P}_y$. Using the bound (4.22) and the inequality $\log \omega \leq \omega - 1$, valid for $\omega > 0$, we obtain

$$d_{\mathrm{TV}}(\mathcal{P}_x, \mathcal{P}_y)^2 \leq \sum_{i=1}^{d} \left(\lambda_i - 2 + \frac{1}{\lambda_i}\right) + \frac{\sqrt{nd}}{r^2} \|x - y\|_x^2 .$$

Using the assumption that $\|x - y\|_x \leq \epsilon r / \left(2(nd)^{1/4}\right)$, and plugging in the bounds (4.23) for the eigenvalues $\{\lambda_i, i \in [d]\}$, we find that

$$\sum_{i=1}^{d} \left(\lambda_i - 2 + \frac{1}{\lambda_i}\right) + \frac{\sqrt{nd}}{r^2} \|x - y\|_x^2 \leq 32\epsilon^2 r^2 + \frac{\epsilon^2}{4}.$$

In asserting this inequality, we have used the facts that according to equation (4.23), for any $i \in [d]$,

$$\lambda_i - 2 + \frac{1}{\lambda_i} = \frac{(\lambda_i - 1)^2}{\lambda_i} \leq 2 \cdot \left(\frac{4\epsilon r}{\sqrt{d}}\right)^2 .$$

Note that for any $r \in [0, 1/12]$ we have that $32r^2 \leq 1/2$. Putting the pieces together yields $d_{\mathrm{TV}}(\mathcal{P}_x, \mathcal{P}_y) \leq \epsilon$, as claimed.

Proof of claim (4.18b)

Note that

$$\mathcal{T}_x(\{x\}) = \mathcal{P}_x(\mathcal{K}^c) + \int_{\mathcal{K}} \left(1 - \min\left\{1, \frac{\rho_z(x)}{\rho_x(z)}\right\}\right) \rho_x(z) dz, \tag{4.24}$$

where $\mathcal{K}^c$ denotes the complement of $\mathcal{K}$. Consequently, we find that

$$\begin{aligned} d_{\mathrm{TV}}(\mathcal{P}_x, \mathcal{T}_x) &= \frac{1}{2}\left(\mathcal{T}_x(\{x\}) + \int_{\mathbb{R}^d} \rho_x(z) dz - \int_{\mathcal{K}} \min\left\{1, \frac{\rho_z(x)}{\rho_x(z)}\right\} \rho_x(z) dz\right) \\ &= \frac{1}{2}\left(2 - 2\int_{\mathbb{R}^d} \min\left\{1, \frac{\rho_z(x)}{\rho_x(z)}\right\} \rho_x(z) dz + 2\int_{\mathcal{K}^c} \min\left\{1, \frac{\rho_z(x)}{\rho_x(z)}\right\} \rho_x(z) dz\right) \\ &\leq \underbrace{\mathcal{P}_x(\mathcal{K}^c)}_{=: S_1} + \underbrace{1 - \mathbb{E}_{z \sim \mathcal{P}_x}\left[\min\left\{1, \frac{\rho_z(x)}{\rho_x(z)}\right\}\right]}_{=: S_2}, \end{aligned} \tag{4.25}$$

Consequently, it suffices to show that both S_1 and S_2 are small, where the probability is taken over the randomness in the proposal z. In particular, we show that $S_1 \leq \epsilon$ and $S_2 \leq 4\epsilon$.

Bounding the term S_1: Since z is multivariate Gaussian with mean x and covariance $\frac{r^2}{\sqrt{nd}} V_x^{-1}$, we can write

$$z \stackrel{d}{=} x + \frac{r}{(nd)^{1/4}} V_x^{-1/2} \xi, \tag{4.26}$$

where $\xi \sim \mathcal{N}(0, \mathbb{I}_d)$ and $\stackrel{d}{=}$ denotes equality in distribution. Using equation (4.26) and definition (4.17b) of $\theta_{V_x,i}$, we obtain the bound

$$\frac{\left(a_i^\top (z-x)\right)^2}{s_{x,i}^2} = \frac{r^2}{(nd)^{\frac{1}{2}}} \left[\frac{a_i^\top V_x^{-1/2} \xi}{s_{x,i}}\right]^2 \stackrel{(i)}{\leq} \frac{r^2}{(nd)^{\frac{1}{2}}} \theta_{V_x,i} \|\xi\|_2^2 \stackrel{(ii)}{\leq} \frac{r^2}{d} \|\xi\|_2^2, \tag{4.27}$$

where step (i) follows from Cauchy-Schwarz inequality, and step (ii) from the bound on $\theta_{V_x,i}$ from Lemma 11(c). Define the events

$$\mathcal{E} := \left\{ \frac{r^2}{d} \|\xi\|_2^2 < 1 \right\} \quad \text{and} \quad \mathcal{E}' := \{ z \in \text{int}(\mathcal{K}) \}.$$

Inequality (4.27) implies that $\mathcal{E} \subseteq \mathcal{E}'$ and hence $\mathbb{P}[\mathcal{E}'] \geq \mathbb{P}[\mathcal{E}]$. Using a standard Gaussian tail bound and noting that $r \leq \frac{1}{1+\sqrt{2/d \log(1/\epsilon)}}$, we obtain $\mathbb{P}[\mathcal{E}] \geq 1-\epsilon$ and whence $\mathbb{P}[\mathcal{E}'] \geq 1-\epsilon$. Thus, we have shown that $\mathbb{P}[z \notin \mathcal{K}] \leq \epsilon$ which implies that $S_1 \leq \epsilon$.

Bounding the term S_2: By Markov's inequality, we have

$$\mathbb{E}_{z \sim \mathcal{P}_x} \left[\min \left\{ 1, \frac{\rho_z(x)}{\rho_x(z)} \right\} \right] \geq \alpha \mathbb{P}[\rho_z(x) \geq \alpha \rho_x(z)] \quad \text{for all } \alpha \in (0,1]. \tag{4.28}$$

By definition (4.12) of ρ_x, we obtain

$$\frac{\rho_z(x)}{\rho_x(z)} = \exp\left(-\frac{\sqrt{nd}}{2r^2} \left(\|z-x\|_z^2 - \|z-x\|_x^2 \right) + \frac{1}{2} \left(\log\det V_z - \log\det V_x \right) \right).$$

The following lemma provides us with useful bounds on the two terms in this expression, valid for any $x \in \text{int}(\mathcal{K})$.

Lemma 14. *For any $\epsilon \in (0, 1/15]$ and $r \in (0, f(\epsilon)]$, we have*

$$\mathbb{P}_{z \sim \mathcal{P}_x} \left[\frac{1}{2} \log\det V_z - \frac{1}{2} \log\det V_x \geq -\epsilon \right] \geq 1-\epsilon, \quad \textit{and} \tag{4.29a}$$

$$\mathbb{P}_{z \sim \mathcal{P}_x} \left[\|z-x\|_z^2 - \|z-x\|_x^2 \leq 2\epsilon \frac{r^2}{\sqrt{nd}} \right] \geq 1-\epsilon. \tag{4.29b}$$

See Appendix B.3 for the proof of this claim.

Using Lemma 14, we now complete the proof. For $r \leq f(\epsilon)$, we obtain

$$\frac{\rho_z(x)}{\rho_x(z)} \geq \exp(-2\epsilon) \geq 1 - 2\epsilon$$

with probability at least $1 - 2\epsilon$. Substituting $\alpha = 1 - 2\epsilon$ in inequality (4.28) yields that $S_2 \leq 4\epsilon$, as claimed.

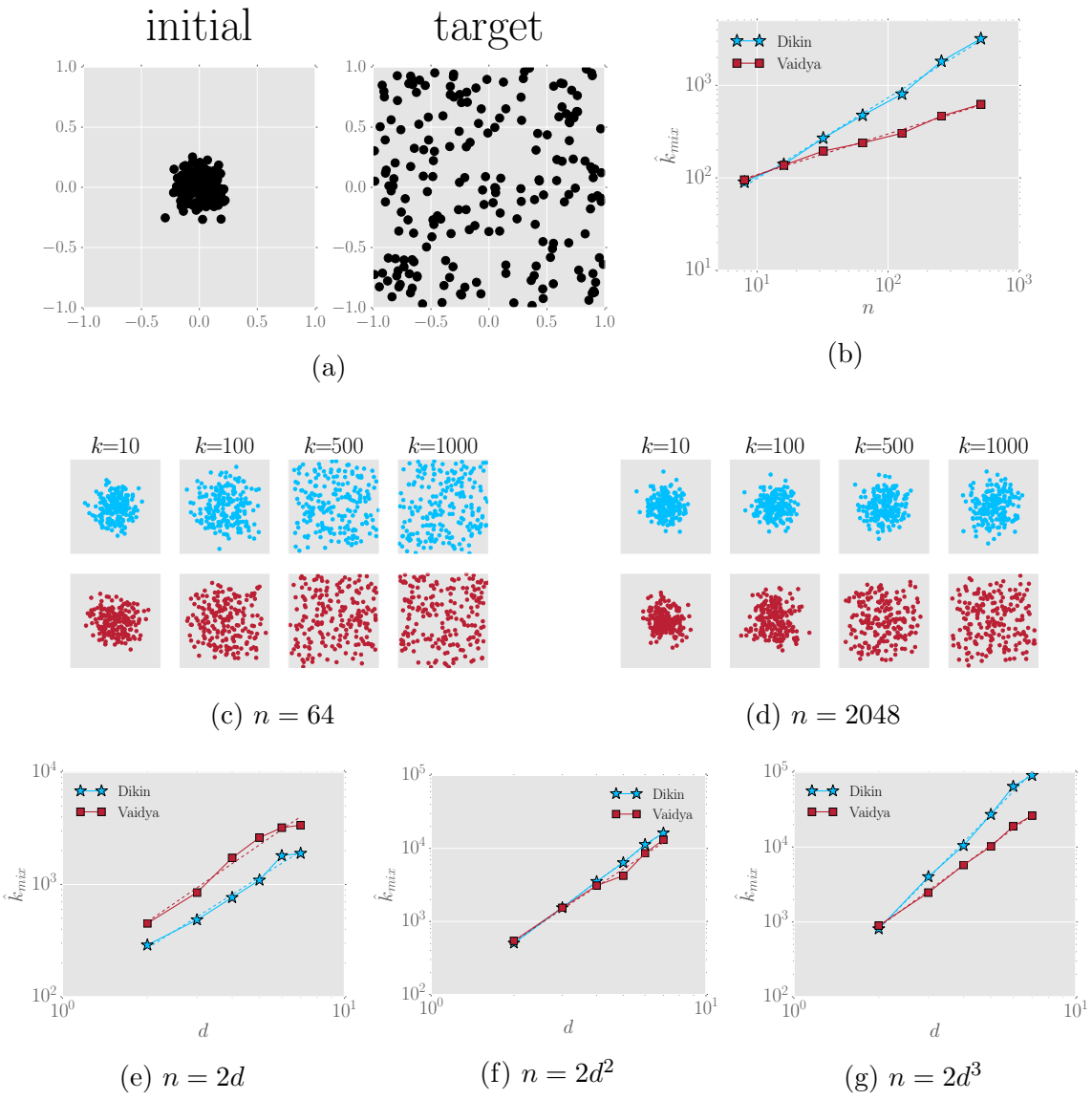
4.6 Summary

In this chapter, we focused on improving mixing rate of MCMC sampling algorithms for polytopes by building on the advancements in the field of interior point methods. We proposed and analyzed two different barrier based MCMC sampling algorithms for polytopes that outperforms the existing sampling algorithms like the ball walk, the hit-and-run and the Dikin walk for a large class of polytopes. We provably demonstrated the fast mixing of the Vaidya walk, $O\left(n^{0.5}d^{1.5}\right)$ and the John walk, $O\left(d^{2.5}\text{poly-log}(n/d)\right)$ from a warm start. Our numerical experiments, albeit simple, corroborated with our theoretical claims: the Vaidya walk mixes at least as fast the Dikin walk and significantly faster when the number of constraints is quite large compared to the dimension of the underlying space. For the John walk, the logarithmic factors were dominant in all our experiments and thereby we deemed the result of importance only for set-ups with polytopes in very high dimensions with number of constraints overwhelmingly larger than the dimensions. Besides, proving the mixing time guarantees for the improved John walk (Conjecture 1) is still an open question.

[141] analyzed a generalized version of the Dikin walk for arbitrary convex sets equipped with self-concordant barrier. From his results, we were able to derive mixing time bounds of $O\left(nd^4\right)$ and $O\left(d^5\text{poly-log}(n/d)\right)$ from a warm start for the Vaidya walk and the John walk respectively. Our proof takes advantage of the specific structure of the Vaidya and John walk, resulting a better mixing rate upper bound the the general analysis provided by [141].

While we have mainly focused on sampling algorithms on polytopes, the idea of using logarithmic barrier to guide sampling can be extended to more general convex sets. The self-concordance property of the logarithmic barrier for polytopes is extended by [6] to more general convex sets defined by semidefinite constraints, namely, linear matrix inequality (LMI) constraints. Moreover, [141] showed that for a convex set in $\mathbb{R}^d$ defined by n LMI constraints and equipped with the log-determinant barrier—the semidefinite analog of the logarithmic barrier for polytopes—the mixing time of the Dikin walk from a warm start is $O\left(nd^2\right)$. It is possible that an appropriate Vaidya walk on such sets would have a speed-up over the Dikin walk. [142] used the Dikin walk to generate samples from time varying log-concave distributions with appropriate scaling

of the radius for different class of distributions. We believe that suitable adaptations of the Vaidya and John walks for such cases would provide significant gains.

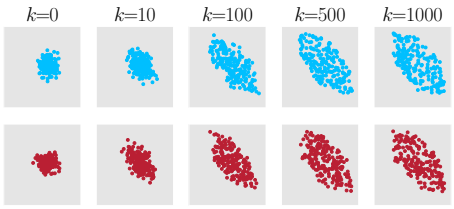


(a) (b)

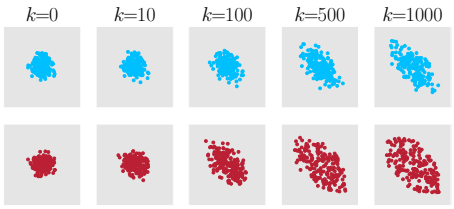
(c) $n = 64$ (d) $n = 2048$

(e) $n = 2d$ (f) $n = 2d^2$ (g) $n = 2d^3$

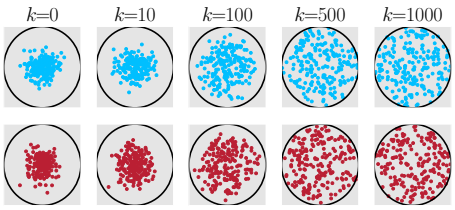
Figure 4.3. Comparison of the Dikin and Vaidya walks on the polytope $\mathcal{K} = [-1, 1]^2$. **(a)** Samples from the initial distribution $\mu_0 = \mathcal{N}(0, 0.04\,\mathbb{I}_2)$ and the uniform distribution on $[-1, 1]^2$. **(b)** Log-log plot of $\hat{k}_{\text{mix}}$ (4.14) versus the number of constraints (n) for a fixed dimension $d = 2$. **(c, d)** Empirical distribution of the samples for the Dikin walk (blue/top rows) and the Vaidya walk (red/bottom rows) for different values of n at iteration $k = 10, 100, 500$ and 1000. **(e, f, g)** Log-log plot of $\hat{k}_{\text{mix}}$ vs the dimension d, for $n \in \{2d, 2d^2, 2d^3\}$ for $d \in \{2, 3, 4, 5, 6, 7\}$. The exponents from these plots are summarized in Table 4.2. Note that increasing the number of constraints n has more profound effect on the Dikin walk in almost all the cases.



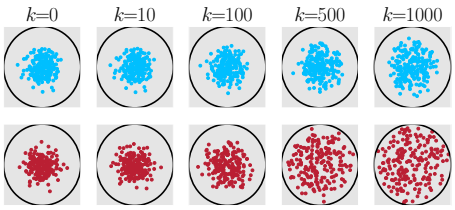
(a) $n = 64$



(b) $n = 2048$

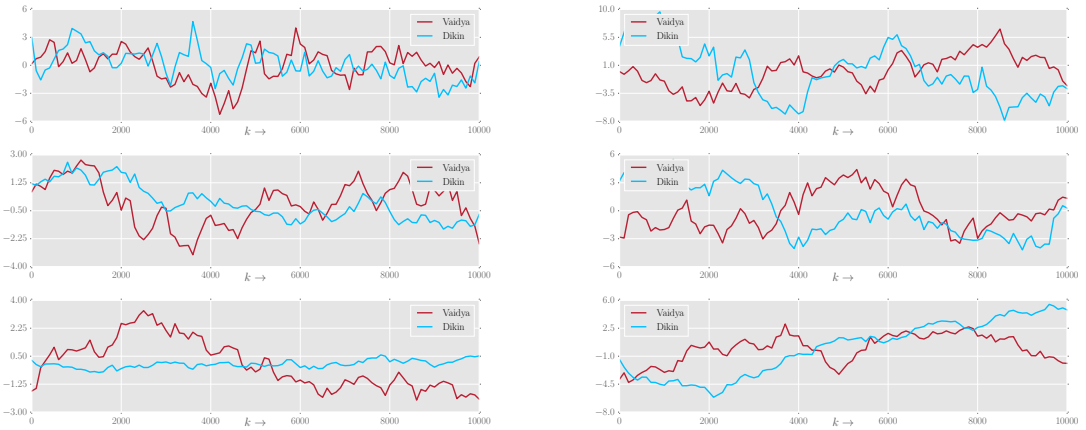


(c) $n = 64$



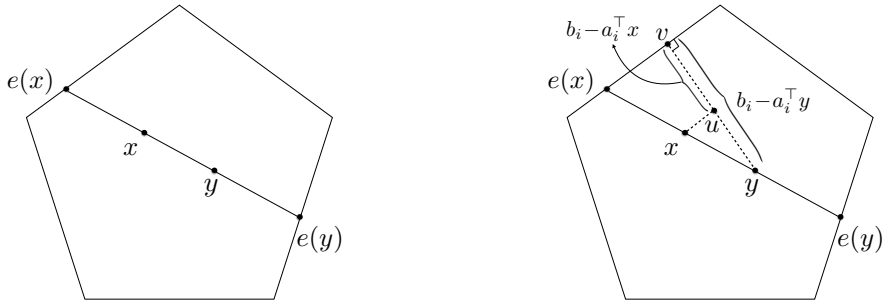
(d) $n = 2048$

Figure 4.4. Empirical distribution of the samples from 200 runs for the Dikin walk (blue/top rows) and the Vaidya walk (red/bottom rows) at different iterations k. The 2-dimensional polytopes considered are: **(a, b)** random polytopes with n-constraints, and **(c, d)** regular n-polygons inscribed in the unit circle. For both sets of cases, we observe that higher n slows down the walks, with visibly more effect on the Dikin walk compared to the Vaidya walk.



(a) $d = 10$ (b) $d = 50$

Figure 4.5. Sequential plots of a one-dimensional random projection of the samples on the hyperbox $\mathcal{K} = [-1, 1]^d$, defined by n constraints. Each plot corresponds to one long run of the Dikin and Vaidya walks, and the projection is taken in a direction chosen randomly from the sphere. **(a)** Plots for $d = 10$ and $n \in \{20, 640, 5120\}$. **(b)** Plots for $d = 50$ and $n \in \{100, 400, 1600\}$. Relative to the Dikin walk, the Vaidya walk has a more jagged plot for pairs (n, d) in which the ratio n/d is relatively large: for instance, see the plots corresponding to $(n, d) = (640, 10)$ and $(5120, 10)$. The same claim cannot be made for pairs (n, d) for which the ratio n/d is relatively small; e.g., the plot with $(n, d) = (20, 10)$. These observations are consistent with our results that the Vaidya walk mixes more quickly by a factor of order $\mathcal{O}(\sqrt{n/d})$ over the Dikin walk.



(a) (b)

Figure 4.6. Polytope $\mathcal{K} = \{x \in \mathbb{R}^d | Ax \leq b\}$. (a) The points $e(x)$ and $e(y)$ denote the intersection points of the chord joining x and y with $\mathcal{K}$ such that $e(x), x, y, e(y)$ are in order. (b) A geometric illustration of the argument (4.20). It is straightforward to observe that $\|x - y\|_2 / \|e(x) - x\|_2 = \|u - y\|_2 / \|u - v\|_2 = \left|a_i^\top (y - x)\right| / \left(b_i - a_i^\top x\right)$.

Part III

Stabilty and learning aspects

Chapter 5

Stability and convergence trade-off

In this chapter of the thesis, we study the interplay between algorithmic stability and convergence rate of iterative machine learning algorithms. For an iterative machine learning algorithm, algorithmic stability controls the generalization error of the algorithm, while convergence rate controls the training error. The statistical learning community has a rich history investigating convergence and stability separately. However, how these two quantities trade off with each other remain open.

We show that for any iterative algorithm at any iteration, the overall performance is lower bounded by the minimax statistical error over an appropriately chosen loss function class. This implies an important trade-off between convergence and stability of the algorithm – a faster converging algorithm has to be less stable, and vice versa. As a direct consequence of this fundamental trade-off, new convergence lower bounds can be derived for classes of algorithms constrained with different stability bounds. When the loss function is convex (or strongly convex) and smooth, we discuss the stability upper bounds of gradient descent (GD) and stochastic gradient descent and their variants with decreasing step sizes. For Nesterov's accelerated gradient descent (NAG) and heavy ball method (HB), we provide stability upper bounds for the quadratic loss function. Applying existing stability upper bounds for the gradient methods in our trade-off framework, we obtain lower bounds matching the well-established convergence upper bounds up to constants for these algorithms and conjecture similar lower bounds for NAG and HB.

In particular, general MCMC sampling algorithms are naturally iterative algorithms. Even though point-estimation is not the typical use of sampling algorithms, but we can still apply them to compare with optimization algorithms. When used for a convex point-estimation problem, we show that Langevin algorithm can not converge faster than gradient descent algorithm because it is more stable.

5.1 Introduction

For different supervised learning algorithms ranging from classical linear regression, logistic regression, boosting, to modern large-scale deep networks, the overall performance or expected excess risk can always be decomposed into two parts: the empirical error (or the training error) and the generalization error (characterizing the discrepancy between the test error and the training error). A central theme in machine learning is to find an appropriate balance between empirical error and generalization error, because improperly emphasizing one over the other typically results in either overfitting or underfitting. Specifically, in the context of supervised learning models trained by iterative optimization algorithms, the empirical error at each iteration is commonly controlled by convergence rate analysis, and the generalization error can be handled by algorithmic stability analysis [50, 20].

Convergence rate of an algorithm portrays how fast the optimization error decreases as the number of iterations grows. Recent years have witnessed a rapid advance on convergence rates analysis of specific optimization methods for a particular class of loss functions that they are optimizing over. In fact, such analysis has been carried out for many gradient methods, including gradient descent (GD), Nesterov accelerated gradient descent (NAG), stochastic gradient descent (SGD), stochastic gradient Langevin dynamics (SGLD) for convex, strongly convex, or even non-convex functions (see e.g. [21, 24, 149, 91, 158]). However, until the optimization error and generalization error of these algorithms are analyzed together, it is not clear whether the fastest converging optimization algorithm is the best for learning.

On the other hand, algorithmic stability [50, 20] in learning problems has been introduced as an alternative way to control generalization error instead of uniform convergence results such as classical VC-theory [192] and Rademacher complexity [11]. The stability concept has an intuitive appeal: an algorithm is stable if it is robust to small perturbations in the composition of the learning data set. Recently it has been shown that algorithmic stability is well suited for controlling generalization error of stochastic gradient methods [75], as well as stochastic gradient Langevin dynamics algorithm [138].

While most previous papers study convergence rate and the algorithmic stability of an optimization algorithm separately, a natural question arises: What is the relationship or trade-off between the convergence rate and the algorithmic stability of an iterative algorithm? Is it possible to design an algorithm that converges the fastest and at the same time most stable? If not, is there any fundamental limit on the trade-off between the two quantities so that a fast algorithm has to be unstable?

This chapter shows that there is a fundamental limit on the trade-off. That is, for any iterative algorithms, at any time step, the sum of optimization error and stability is lower bounded by the minimax statistical error over a given loss function class. Therefore, a fast converging algorithm can not be too stable, and a stable algorithm can not converge too fast. This framework therefore provides a new criterion for comparing optimization algorithms by considering jointly convergence rate and algorithm stability.

As a consequence, our framework can be immediately applied to provide a new class of convergence lower bounds for algorithms with different stability rates.

In particular, we focus on two settings where the loss functions are either convex smooth or strongly convex smooth. In the first setting, we discuss the stability upper bounds of gradient descent (GD), stochastic gradient descent (SGD) and their variants with decreasing step sizes. New stability upper bounds are provided for Nesterov's accelerated gradient descent (NAG) and the heavy ball method (HB) under quadratic loss, and we conjecture these upper bounds still hold for the general convex smooth losses. Applying the stability upper bounds for GD and SGD in our trade-off framework, we obtain the convergence lower bounds for them that match the known convergence upper bounds up to constants. Considering jointly convergence rate and algorithm stability for NAG and GD, the trade-off shows that NAG must be less stable than GD even though it converges faster than GD. In the second setting where the loss functions are strongly convex and smooth, we also provide stability upper bound and deduce the convergence lower bound results for GD and NAG via our trade-off framework. Finally, simulations are conducted to show that the stability bounds established have the correct rates as a function of n and iteration T. These bounds are demonstrated to be particularly useful in large scale learning settings for understanding the overall performance of an algorithm than the classical uniform convergence bounds because the stability bounds capture better generalization errors at early iterations of these algorithms.

Past work on algorithmic stability: The first quantitative results that focus on generalization error via algorithmic stability date back 1970s [170, 50]. This line of research was further developed by Bousquet and Elisseeff [20] to provide guarantees for general supervised learning algorithms and insights for the practice of regularized algorithms. It remains unclear, however, what is the algorithmic stability of general iterative optimization algorithms. Recently, to show the effectiveness of commonly used optimization algorithms in many large-scale learning problems, algorithmic stability has been established for stochastic gradient methods [75], stochastic gradient Langevin dynamics [138], as well as for any algorithm in situations where global minima are approximately achieved [32].

Past work on lower bounds on convergence rate Given the importance of efficient optimization methods, many papers have been devoted to understanding the fundamental computational limits of convex optimization. Those lower bounds typically focus on a specific class of algorithms. A classical line of research has been focused on first-order algorithms where only first-order information (i.e. gradients) can be queried through oracle model; see the book [21], the monograph [24] and references therein for further details. For convex functions, the first lower bound argument given in [147] applies to first-order algorithms whose current iterate lies in the linear span

of previous gradients. It has been later extended to any deterministic, then stochastic first-order algorithm [2, 199].

Organization: The rest of the chapter is organized as follows: In Section 5.2, we set up the necessary backgrounds on the classical excess risk decomposition and introduce the optimization error (or computational bias) and generalization error trade-off. In Section 5.3, we provide the main theorem on the trade-off between convergence rate (as an upper bound on optimization error) and algorithmic stability (as an upper bound on generalization error). In Section 5.4, we establish uniform stability bounds for several gradient methods and show that our main theorem applies to these algorithms to obtain their convergence lower bounds. In Section 5.6, we first provide simulation results validating the correct rates as a function of sample size n and iteration number T of the stability bounds we established, and then illustrate via a simulated logistic regression problem that our stability bounds reflect the generalization errors better than the simple uniform convergence bounds for GD and NAG.

5.2 Preliminaries

In this section, we set up the necessary backgrounds on excess risk decomposition and convex optimization. Using classical excess risk decomposition, we introduce the expected optimization error and generalization error trade-off which will be crucial to state our main result in the next section.

5.2.1 Excess risk decomposition

Throughout this chapter, we consider the standard setting of supervised learning. Suppose that we are given n samples $S = (z_1, ..., z_n)$, each lying in some space $\mathcal{Z}$ and drawn i.i.d. according to a distribution $P \in \mathcal{P}$. The standard decision-theoretic approach is to estimate a parameter $\theta \in \mathbb{R}^d$ by minimizing a loss function of the form $l(\theta; z)$, which measures the fit between the model indexed by the parameter $\theta \in \Omega \subset \mathbb{R}^d$ and the sample $z \in \mathcal{Z}$.

Given the collection S of n samples and a loss function l, the principle of empirical risk minimization is based on the objective function

$$R_S(\theta) \equiv \frac{1}{n} \sum_{i=1}^{n} l(\theta; z_i).$$

This empirical risk above serves as a sample-average proxy for the population risk

$$R(\theta) \equiv \mathbb{E}_{z \sim P}[l(\theta; z)].$$

We denote by $\hat{\theta}$ an estimator computed from sample S. The statistical question is how to bound the excess risk, measured in terms of the difference between the population risk and the minimal risk over the entire parameter space Ω,

$$\delta R(\hat{\theta}) \equiv R(\hat{\theta}) - \inf_{\theta \in \Omega} R(\theta).$$

In most of our analysis, $\hat{\theta}$ is the output of an optimization algorithm at a particular iteration T based on sample S. We further denote $\tilde{\theta}$ an empirical risk minimizer. Note that $\hat{\theta}$ and $\tilde{\theta}$ are in general not the same estimator.

For simplicity, we assume that there exists some $\theta_0 \in \Omega$ such that $R(\theta_0) = \inf_{\theta \in \Omega} R(\theta)$.[1]

Controlling the excess risk of the estimator $\hat{\theta}$ is usually done by decomposing it into three terms as follows:

$$\delta R(\hat{\theta}) = \underbrace{R(\hat{\theta}) - R_S(\hat{\theta})}_{T_1} + \underbrace{R_S(\hat{\theta}) - R_S(\theta_0)}_{T_2} + \underbrace{R_S(\theta_0) - R(\theta_0)}_{T_3}.$$

Term T_1 is the generalization error of the model $\hat{\theta}$. Term T_2 is the empirical risk difference between the model $\hat{\theta}$ and the population risk minimizer θ_0. Term T_3 is the generalization error of θ_0.

Taking expectation on the previous decomposition and noticing that $\mathbb{E}_S[T_3] = 0$, we obtain first a decomposition of the expected excess risk and then an upper bound:

$$\begin{aligned}
\mathbb{E}_S[\delta R(\hat{\theta})] &= \mathbb{E}_S[R(\hat{\theta}) - R_S(\hat{\theta})] + \mathbb{E}_S[R_S(\hat{\theta}) - R_S(\theta_0)] + 0 \\
&= \underbrace{\mathbb{E}_S[R(\hat{\theta}) - R_S(\hat{\theta})]}_{\mathcal{E}_{\text{gen}}} + \underbrace{\mathbb{E}_S[R_S(\hat{\theta}) - R_S(\tilde{\theta})]}_{\mathcal{E}_{\text{opt}}} + \underbrace{\mathbb{E}_S[R_S(\tilde{\theta}) - R_S(\theta_0)]}_{\leq 0} \\
&\leq \underbrace{\mathbb{E}_S[R(\hat{\theta}) - R_S(\hat{\theta})]}_{\mathcal{E}_{\text{gen}}} + \underbrace{\mathbb{E}_S[R_S(\hat{\theta}) - R_S(\tilde{\theta})]}_{\mathcal{E}_{\text{opt}}}.
\end{aligned}$$

The last inequality follows from the fact that $\tilde{\theta}$ is the empirical risk minimizer. Hence, the expected excess risk is upper bounded by the sum of expected generalization error and the *expected optimization error* or *computational bias* $\mathbb{E}_S[R_S(\hat{\theta}) - R_S(\tilde{\theta})]$. We formally define these two quantities indexed by the estimator $\hat{\theta}$, loss function l, data distribution P and sample size n to be

$$\mathcal{E}_{\text{gen}}(\hat{\theta}, l, P, n) \equiv \mathbb{E}_{S \sim P^n}\left[R(\hat{\theta}) - R_S(\hat{\theta})\right],$$

and

$$\mathcal{E}_{\text{opt}}(\hat{\theta}, l, P, n) \equiv \mathbb{E}_{S \sim P^n}\left[R_S(\hat{\theta}) - R_S(\tilde{\theta})\right].$$

[1] If the infimum is not achieved within Ω (for example Ω is an open set), we can choose some θ_0 where this equality holds up to some arbitrarily small error.

Making the optimization error appear in the decomposition is useful for analyzing optimization algorithms in an iterative manner. As noted in Bousquet and Bottou [19], introducing optimization error allows to analyze algorithms doing approximate optimization. However, our framework is different to that introduced before. We control the generalization error via iteration-dependent algorithmic stability instead of directly invoking uniform convergence results. As we are going to show, for most iterative optimization algorithms, upper bounding the generalization error by a simple uniform convergence is often loose and algorithmic stability can serve as a tighter bound.

5.2.2 Algorithmic Stability

Many forms of algorithmic stability have been introduced to characterize generalization error [20, 102]. For the purpose of this chapter, we are only interested in the *uniform stability* notion introduced by Bousquet and Elisseeff [20].

Definition 3. *An algorithm, which outputs a model $\hat{\theta}_S$ for sample S, is* ***ϵ-uniform stable*** *if for all $k \in \{1, ..., n\}$, for all data sample pair $S = (z_1, ..., z_k, ..., z_n)$ and $S' = (z_1, ..., z'_k, ..., z_n)$, each z_i or z'_k is i.i.d sampled from P, we have*

$$\sup_{z \in \mathcal{Z}} \left| l(\hat{\theta}_S; z) - l(\hat{\theta}_{S'}; z) \right| \leq \epsilon. \tag{5.1}$$

As we did for the generalization error, we use $\mathcal{E}_{\text{stab}}(\hat{\theta}, l, P, n)$ to denote the uniform stability of an algorithm $\hat{\theta}$.

A stable algorithm has the property that removing one element in its learning data set does not change much of its outcome. Such a data perturbation scheme is closely related to Jackknife in statistics [61]. One can further show that uniform stability implies expected generalization [20] . For completeness, we reformulate this property in the following lemma.

Lemma 15. *An algorithm, which outputs a model $\hat{\theta}_S$ for sample S, is ϵ-uniformly stable, then its expected generalization error is bounded as follows,*

$$\left| \mathbb{E}_S \left[R(\hat{\theta}_S) - R_S(\hat{\theta}_S) \right] \right| \leq \epsilon.$$

Lemma 15 implies that $\mathcal{E}_{\text{gen}}(\hat{\theta}, l, P, n) \leq \mathcal{E}_{\text{stab}}(\hat{\theta}, l, P, n)$. The proof provided by [20] relies on a symmetrization argument and makes use of the i.i.d assumptions of samples in S. Combining the expected excess risk decomposition in previous section, we conclude that the sum of uniform stability and expected optimization error (or computational bias) constitutes an upper bound for the expected excess risk,

$$\mathbb{E}_{S \sim P^n}[\delta R(\hat{\theta}_S)] \leq \mathcal{E}_{\text{stab}}(\hat{\theta}, l, P, n) + \mathcal{E}_{\text{opt}}(\hat{\theta}, l, P, n). \tag{5.2}$$

Note that the result is stated for a fixed loss function l and a fixed data distribution P. Equation (5.2) is a key inequality for our analysis. Not only it provides a way to upper

bound the expected excess risk without uniform convergence results, but also it makes the connection between the statistical excess risk and the optimization convergence rate (or computational bias). This can also be seen as reminiscent of the bias-variance trade-off of an algorithm in a computational sense since stability serves as a computational variability term and optimization error as a computational bias term.

5.2.3 Convex optimization settings

Throughout the chapter, we focus on two types of loss functions: The first type of loss function $l(\cdot, z)$ is m-strongly convex and L-smooth for every z; The second type of loss function $l(\cdot, z)$ is convex and L-smooth for every z. We will also make use of the M-Lipschitz condition. More technical details about convex optimization and relevant results are deferred to Appendix C.1.

5.3 Trade-off between stability and convergence rate

In this section, we introduce the trade-off between stability and convergence rate via excess risk decomposition under two settings of loss functions mentioned in the previous section: the convex smooth setting and the strongly convex smooth setting. We show that for any iterative algorithm, at any time step, the sum of optimization error and stability is lower bounded by the minimax statistical error over a given loss function class. Thus algorithms sharing the same stability upper bound can be grouped to obtain convergence rate lower bounds. This provides a new class of convergence lower bounds for algorithms with different stability bounds.

We are interested in distribution independent stability and convergence where we take supremum of these two quantities over distributions and losses. For a fixed iteration algorithm that outputs $\hat{\theta}$ at iteration T, we define its uniform stability and optimization error as follows,

$$\mathcal{E}^{\hat{\theta}}_{\text{stab}}(T, n, \mathcal{L}) \equiv \sup_{l \in \mathcal{L}, P \in \mathcal{P}} \mathcal{E}_{\text{stab}}(\hat{\theta}_T, l, P, n),$$
$$\mathcal{E}^{\hat{\theta}}_{\text{opt}}(T, n, \mathcal{L}) \equiv \sup_{l \in \mathcal{L}, P \in \mathcal{P}} \mathcal{E}_{\text{opt}}(\hat{\theta}_T, l, P, n).$$

Note that in this chapter, the supremum is taken over the class of all loss functions $\mathcal{L}$ under either of the two settings considered (convex smooth and strongly convex smooth settings).

5.3.1 Trade-off in the convex smooth setting

Before we state the main theorem, we first define the loss function class of interest in this section. We define the class of all convex smooth loss functions as follows,

$$\mathcal{L}_{\mathrm{c}} = \{l : \Omega \times \mathcal{Z} \to \mathbb{R} | l \text{ is convex, } L\text{-smooth}, |\Omega| = R\}.$$

Here $|\Omega|$ is defined as $\sup_{\theta\in\Omega} \|\theta\|_2$. In the convex smooth setting, we have the following lower bound on the sum of stability and convergence rate.

Theorem 7. *Suppose an iterative algorithm outputs $\hat{\theta}_T$ at iteration T on an empirical loss built upon a loss $l \in \mathcal{L}_c$ and an i.i.d. sample S of size n, and it has uniform stability $\mathcal{E}_{stab}(T, n, \mathcal{L}_c)$ and optimization error $\mathcal{E}_{opt}(T, n, \mathcal{L}_c)$, then there exists a universal constant $C_1 > 0$ such that,*

$$\mathcal{E}^{\hat{\theta}}_{stab}(T, n, \mathcal{L}_c) + \mathcal{E}^{\hat{\theta}}_{opt}(T, n, \mathcal{L}_c) \geq \inf_{\tilde{\theta}} \sup_{P\in\mathcal{P}} \mathbb{E}_{S\sim P^n}[\delta R(\tilde{\theta})] \geq \frac{R^2 L}{C_1\sqrt{n}}$$

The first inequality of Theorem 7 is a simple outcome of the empirical risk decomposition in Equation (5.2). This first inequality is not tied to the convex smooth setting and can generalize to a wide class of optimization algorithms. The second inequality is based on an adaptation of the classical [104]'s method for minimax estimation lower bound to the convex smooth loss function class. Further, if we know $\mathcal{E}^{\hat{\theta}}_{\mathrm{stab}}(T, n, \mathcal{L}_{\mathrm{c}})$ precisely, we can obtain an immediate corollary that provide convergence lower bound for stable optimization algorithms.

Corollary 5. *Under conditions in Theorem 7, if an algorithms has uniform stability*

$$\mathcal{E}^{\hat{\theta}}_{stab}(T, n, \mathcal{L}_c) \leq \frac{s(T)}{n},$$

with s a divergent function of T, i.e.

$$s(T) \to \infty, \text{ as } T \to \infty,$$

then there exists a universal constant $C_2 > 0$, a sample size n_0 and an iteration number $T_0 \geq 1$, such that for $T \geq T_0$, its convergence rate is lower bounded as follows,

$$\mathcal{E}^{\hat{\theta}}_{opt}(T, n_0, \mathcal{L}_c) \geq \frac{R^4 L^2}{C_2 s(T)}.$$

Even though Theorem 7 is valid for any pair of (T, n), Corollary 5 requires to choose a specific sample size n_0 in construction. However, under the assumption that the optimization algorithm has convergence rate independent of the sample size (i.e. $\mathcal{E}^{\hat{\theta}}_{\mathrm{opt}}(T, n, \mathcal{L}_{\mathrm{c}})$ is not a function of n), we can obtain via Corollary 5 a convergence lower bound that is comparable to the lower bounds in the convex optimization literature.

We remark that this assumption is satisfied for commonly-used optimization algorithms such as GD and NAG.

Theorem 7 and Corollary 5 provide the trade-off between stability and optimization convergence rate. All iterative optimization methods that are algorithmic uniform stable can not converge too fast. This motivates the idea of grouping optimization methods with their algorithmic stability. Optimization methods that share the same algorithmic stability would have the same optimization lower bound. The proof of Theorem 7 is provided in Appendix 5.5.1 and that of Corollary 5 in Appendix 5.5.2.

5.3.2 Trade-off in the strongly convex smooth setting

Similar to the convex smooth setting, we define the class of all strongly convex smooth loss functions as follows,

$$\mathcal{L}_{\text{sc}} = \{l : \Omega \times \mathcal{Z} \to \mathbb{R} | l \text{ is } m\text{-strongly convex, } L\text{-smooth}, |\Omega| = R\}.$$

In the strongly convex smooth setting, we have the following lower bound on the sum of stability and convergence rate.

Theorem 8. *Suppose an iterative algorithm outputs $\hat{\theta}_T$ at iteration T on an empirical loss built upon a loss $l \in \mathcal{L}_{sc}$ and an i.i.d. sample S of size n, and it has uniformly stability $\mathcal{E}^{\hat{\theta}}_{stab}(T, n, \mathcal{L}_{sc})$ and has optimization error $\mathcal{E}^{\hat{\theta}}_{opt}(T, n, \mathcal{L}_{sc})$, then there exists a universal constant C_3 such that*

$$\mathcal{E}^{\hat{\theta}}_{stab}(T, n, \mathcal{L}_{sc}) + \mathcal{E}^{\hat{\theta}}_{opt}(T, n, \mathcal{L}_{sc}) \geq \inf_{\tilde{\theta}} \sup_{P \in \mathcal{P}} \mathbb{E}_{S \sim P^n}[\delta R(\tilde{\theta})] \geq \frac{R^2 L}{C_3 n}.$$

The trade-off in the strongly convex smooth setting is similar to that of convex smooth setting, except that the minimax estimation rate is of order $O(\frac{1}{n})$ instead of $O(\frac{1}{\sqrt{n}})$. Theorem 8 provides the trade-off between stability and optimization convergence rate in the strongly convex setting. Note that a similar corollary like Corollary 5. The proof of Theorem 8 is provided in Appendix 5.5.3.

5.4 Stability and implications for convergence lower bounds

This section is devoted to establishing stability bounds of popular first order optimization algorithms and showing that our main theorem can be applied to these algorithms to obtain their convergence lower bounds. In particular, Subsection 5.4.1 establishes uniform stability for first order iterative methods in the convex smooth setting and Subsection 5.4.2 discusses the consequence after applying Theorem 7 to various optimization algorithms. Subsection 5.4.3 provides uniform stability for first order iterative algorithms in the strongly convex smooth setting and Subsection 5.4.4 discusses the consequence after applying Theorem 8 to GD and NAG.

The goal of proving uniform stability for iteration T is to bound the difference

$$\left|l(\hat{\theta}_{S,T}; z) - l(\hat{\theta}_{S',T}; z)\right|$$

for the sample $S = (z_1, \ldots, z_k, \ldots, z_n)$ and the perturbed one $S' = (z_1, \ldots, z'_k, \ldots, z_n)$, uniformly for every $z \in \mathcal{Z}$. $z_1, \ldots, z_k, \ldots, z_n$ and z'_k are drawn i.i.d from a distribution P. Here $\hat{\theta}_{S,T}$ denotes the output model of our optimization algorithm at iteration T based on sample S. The optimization algorithm is applied on a pair of data samples S, S' to get two sequences of successive models $\hat{\theta}_{S,0}, \hat{\theta}_{S,0}, \ldots, \hat{\theta}_{S,T}$ and $\hat{\theta}_{S',0}, \hat{\theta}_{S',1}, \ldots, \hat{\theta}_{S',T}$. For simplicity, we use $\hat{\theta}_t$ to denote $\hat{\theta}_{S,t}$ and $\hat{\theta}'_t$ for $\hat{\theta}_{S',t}$. We first bound the model estimate difference $\left\|\hat{\theta}_t - \hat{\theta}'_t\right\|_2$, then use the M-Lipschitz condition of l to prove stability.

Recall that the empirical loss function for data sample $S = (z_1, \ldots, z_n)$ is

$$R_S(\theta) \equiv \frac{1}{n}\sum_{j=1}^{n} l(\theta; z_j) = \frac{1}{n}\sum_{j=1}^{n} f_j(\theta).$$

where we have replaced $l(\theta; z_j)$ with $f_j(\theta)$ to improve readability. On the other hand, the empirical loss function for the perturbed sample $S' = (z_1, \ldots, z'_k, \ldots, z_n)$ is

$$R_{S'}(\theta) = \left[\frac{1}{n}\sum_{i=1, i\neq k}^{n} l(\theta; z_i)\right] + \frac{1}{n} l(\theta; z'_k) = \left[\frac{1}{n}\sum_{i=1, i\neq k}^{n} f_i(\theta)\right] + \frac{1}{n} f'_k(\theta).$$

Remark that the two empirical loss functions only differ on one term that is proportional to the inverse of sample size n.

5.4.1 Stability in the convex smooth setting

We establish uniform stability for gradient descent, stochastic gradient descent, Nesterov accelerated gradient method and heavy ball method with fixed momentum parameter when the loss function is convex smooth.

Gradient descent (GD)

The gradient descent algorithm is an iterative method for optimization, which uses the full gradient at each iteration (See book by [21]). Given a convex smooth objective F, GD starts at some initial point $\theta_0 \in \Omega$, and iterates with the following recursion

$$\theta_{t+1} = \theta_t - \eta \nabla F(\theta_t), \ t = 1, 2, \cdots,$$

where η is the step-size. Typically, one would choose fixed $\eta \leq \frac{1}{L}$ to ensure convergence [21]. In the empirical risk minimization setting, the objective F of the optimization is either R_S or $R_{S'}$.

Theorem 9. *Given a data distribution P, under the assumption that $l(\cdot, z)$ is a convex, M-Lipschitz and L-smooth function for every $z \in \mathcal{Z}$, the gradient method with constant step-size $\eta \leq \frac{1}{L}$ on the empirical risk R_S with sample size n, which outputs $\hat{\theta}_T$ at iteration T, has the following uniform stability bound for all $T \geq 1$,*

$$\mathcal{E}_{stab}^{GD}(\hat{\theta}_T, l, P, n) \leq \frac{2\eta M^2 T}{n}. \tag{5.3}$$

We remark that this stability bound does not depend on the exact form of the loss function l and the exact form of the data distribution P. The proof of this theorem is provided in Appendix C.1.1. The key step of our proof is that in such a set-up, the error caused by the difference in empirical loss functions accumulates linearly as the iteration increases. We also show in Appendix C.1.1 that this stability upper bound can be achieved by a linear loss function.

Nesterov accelerated gradient methods (NAG)

The Nesterov's accelerated gradient method attains the optimal convergence rate $O(1/T^2)$ in the smooth non-strongly convex setting under the deterministic first order oracle [148]. Given a convex smooth objective F, starting at some initial point $\theta_0 = w_0 \in \Omega$, NAG uses the following updates,

$$\begin{aligned} \theta_{t+1} &= w_t - \eta \nabla F(w_t), \\ w_{t+1} &= (1 - \gamma_t)\theta_{t+1} + \gamma_t \theta_t, \end{aligned}$$

where $\eta \leq \frac{1}{L}$ is the step-size. The parameter γ_t is defined by the following recursion

$$\lambda_0 = 0, \lambda_t = \frac{1 + \sqrt{1 + 4\lambda_{t-1}^2}}{2}, \text{ and } \gamma_t = \frac{1 - \lambda_t}{\lambda_{t+1}},$$

satisfying $-1 < \gamma_t \leq 0$. We only provide a uniform stability bound for NAG when the empirical risk function is quadratic. We conjecture that the same stability bound holds for general convex smooth functions.

Theorem 10. *Given a data distribution P, under the assumption that $l(\cdot, z)$ is a M-Lipschitz, L-smooth convex quadratic loss function defined on a bounded domain for every $z \in \mathcal{Z}$, Nesterov accelerated gradient method with fixed step-size $\eta \leq \frac{1}{L}$, which outputs $\hat{\theta}_T$ at iteration T, has the following uniform stability bound for all $T \geq 1$,*

$$\mathcal{E}_{stab}^{NAG}(\hat{\theta}_T, l, P, n) \leq \frac{4\eta M^2 T^2}{n}. \tag{5.4}$$

The proof of the theorem is provided in Appendix C.1.2. We also show in Appendix that this stability upper bound is achieved by a linear loss function. Note that unlike the full gradient method and stochastic gradient descent, the stability bound of Nesterov accelerate gradient method depends quadratically on the iteration T. Even though NAG can still have small stability when early stopping is used, its stability grows faster than that of GD at the same iteration.

The heavy ball method with a fixed momentum

The heavy ball method (HB), like NAG, is also a multi-step extension of the gradient descent method [157]. Fixed step-size and fixed momentum parameter heavy ball method has the following updates. For $t \geq 1$,

$$\theta_{t+1} = \theta_t - \eta \nabla F(\theta_t) + \gamma \left(\theta_t - \theta_{t-1}\right),$$

with fixed $\gamma \in [0, 1), \eta \in \left(0, \frac{2(1-\gamma)}{L}\right)$. As for the NAG, we provide only a uniform stability bound for the heavy ball method when the empirical risk function is quadratic. We conjecture that the same stability bound holds for general convex smooth functions.

Theorem 11. *Given a data distribution P, under the assumption that $l(\cdot, z)$ is a M-Lipschitz, L-smooth convex quadratic loss function defined on a bounded domain for every z, the heavy ball method with a fixed step-size $\eta \in \left(0, \frac{(1-\gamma)}{L}\right)$ and a fixed momentum parameter $\gamma \in [0, 1)$, which outputs $\hat{\theta}_T$ at iteration T, has the following uniform stability bound for all $T \geq 1$,*

$$\mathcal{E}_{stab}^{HB,\ fixed}(\hat{\theta}_T, l, P, n) \leq \frac{4\eta M^2 T}{(1 - \sqrt{\gamma})n}. \tag{5.5}$$

The proof of this theorem is provided in Appendix C.1.3. This theorem shows that the Heavy ball method with a fixed step-size and a fixed momentum parameter also uses multi-step gradients, it is more stable than NAG with a stability bound of order $O(T/n)$. This demonstrates that the multi-step setup does not necessarily lead to a similar or worse stability bound than that of NAG.

Other methods with known stability

In this subsection, we restate the stability bounds of some other gradient methods in this subsection for completeness. The stability bounds stated in this subsection are not new, but they will serve as basis of our discussion for their convergence lower bounds implied by Theorem 7 in Subsection 5.4.2.

Stochastic gradient descent (SGD) with fixed or varying step-size The stochastic gradient descent is a randomized iterative algorithm for optimization. Instead of using the full gradient information, it randomly chooses one data sample and updates the parameter estimate according to the gradient on that sample. It starts at some initial point $\theta_0 \in \Omega$, and iterates with the following recursion with i chosen from the set $\{1, ..., n\}$ uniformly at random:

$$\theta_{t+1} = \theta_t - \eta \nabla f_i(\theta_t), t = 1, 2, \ldots$$

Hardt et al. [75] adapted the definition of uniform stability to randomized algorithms and showed that the fixed step-size $\eta \leq \frac{1}{L}$ stochastic gradient descent has a $\frac{2\eta M^2 T}{n}$-uniform stability bound in the convex, M-Lipschitz and L-smooth setting. According to Theorem 3.8 in Hardt et al. [75], we have following restatement in our notation,

$$\mathcal{E}_{\text{stab}}^{\text{SGD, fixed}}(\hat{\theta}_T, l, P, n) \leq \frac{2\eta M^2 T}{n} \tag{5.6}$$

for any convex M-Lipschitz and L-smooth loss function l.

Hardt et al. [75] further considers stochastic gradient descent with decreasing step-sizes $\eta_t = t^{-m}$ and shows that stochastic gradient descent with decreasing step-sizes has $\frac{2\eta M^2 T^{1-m}}{n}$-uniform stability in the same setting.

Stochastic gradient Langevin dynamics (SGLD) Stochastic gradient Langevin dynamics (SGLD) is a popular variant of stochastic gradient descent, where properly scaled isotropic Gaussian noise is added to an unbiased estimate of the gradient at each iteration [67]. Stochastic gradient Langevin dynamics with temperature parameter τ and step-size η_t, starts at some initial point $\theta_0 \in \mathbb{R}^n$, and iterates with the following recursion with i chosen from the set $\{1, ..., n\}$ uniformly at random, and $w \sim \mathcal{N}(0, \mathbb{I}_d)$,

$$\theta_{t+1} = \theta_t - \eta_t \nabla f_i(\theta_t) + \sqrt{\frac{2\eta_t}{\tau}} w.$$

SGLD plays an important role in sampling and optimization. It is proposed as a stochastic discrete version of the Langevin Equation $d\theta_t = -\nabla f(\theta_t)dt + \sqrt{\frac{2}{\tau}} dB_t$, where B_t is the Brownian motion. Recent work by [158] has shown its effective in non-convex learning with optimization and generalization guarantees.

When SGLD is applied to optimization, a decreasing step with $\eta_t = O(\eta_0/t)$ should be used to ensure convergence to local minima. We study this particular step-size setting of SGLD. It has been shown by [138] that SGLD has the following uniform stability for M-Lipschitz convex loss function,

$$O\left(\frac{M}{n}\left(k_0 + M\sqrt{\tau \sum_{t=k_0+1}^{T} \eta_t}\right)\right),$$

where $k_0 = \min\{t | \eta_k \tau M^2 < 1\}$. Plugging in the $O(\eta_0/t)$ step-size, we have that SGLD has a uniform stability bound

$$\mathcal{E}_{\text{stab}}^{\text{SGLD}}(\hat{\theta}_T, l, P, n) \leq O(\frac{M^2 (\tau\eta_0)^{1/2} T^{1/4}}{n}), \tag{5.7}$$

at iteration $T \geq 1$, for any convex M-Lipschitz and L-smooth loss function l. This is an adaptation of the result of Mou et al. [138] in our notation. The goal is to illustrate the trade-off results in the next section.

5.4.2 Consequences for the convergence lower bound in convex smooth setting

In this section, we apply Theorem 7 and Corollary 5 to obtain convergence lower bounds for a variety of first order optimization algorithms mentioned above. Furthermore, we compare the convergence lower bound we obtain with the known convergence upper bound for each of the optimization methods mentioned in the previous section. The known convergence upper bounds mentioned in this section can be found in the optimization textbooks (See [21] or [24]). We also discuss how our lower bounds compare to those obtained from classical oracle model of complexity by [147].

Note that the assumptions in Theorem 7 are slightly different to what we use when we establish stability bounds in the previous section: the former assume bounded domain R while the latter assume M-Lipschitz. To make these two assumptions compatible, in this subsection, we assume that the domain $R = |\Omega|$ is fixed and for all $z \in \mathcal{Z}$, there exists $\theta^* \in \Omega$ such that $\nabla l(\cdot, z) = 0$. Then we have the loss is M-Lipschitz with $M \leq RL$. This is because for any $\theta \in \Omega$,

$$\|\nabla l(\theta, z)\|_2 = \|\nabla l(\theta, z) - \nabla l(\theta^*, z)\|_2 \leq L \|\theta - \theta^*\|_2 \leq RL.$$

In Table 5.1, we summarize all the uniform stability results and the corresponding convergence lower bound under convex smooth setting. While exact constants are provided in the main text, we only show the dependency on iteration number T and sample size n in the table.

Gradient descent

According to Equation (5.3) in Theorem 9, the fixed-step-size full gradient method has $\frac{2\eta(RL)^2T}{n}$-uniform stability. Applying Corollary 5, knowing that its convergence does not depend on n, we obtain that its convergence rate is lower bounded by

$$\mathcal{E}^{\mathrm{GD}}_{\mathrm{opt}}(T, \mathcal{L}_{\mathrm{c}}) \geq \frac{R^2}{2C_2\eta T}. \tag{5.8}$$

It is known (see e.g. [24]) that for f convex an L-smooth on $\mathbb{R}^n$, the full gradient method with step-size $\eta \leq \frac{1}{L}$ satisfies

$$f(\theta_T) - f(\theta^*) \leq \frac{2\|\theta_0 - \theta^*\|^2}{\eta T}.$$

The convergence rate lower bound obtained via our stability trade-off thus matches the known upper bound up to constant factors.

Stochastic gradient descent

According to [75], the fixed step-size stochastic gradient descent also has $\frac{2\eta(RL)^2T}{n}$-uniform stability. Applying Corollary 5, we obtain a convergence rate lower bound of

Method	Uniform stab.	Conv. upper (known)	Conv. lower (ours)
GD, $\eta = 1/L$	$O\left(\frac{T}{n}\right)$	$O\left(\frac{1}{T}\right)$	$O\left(\frac{1}{T}\right)$
NAG*	$O\left(\frac{T^2}{n}\right)$	$O\left(\frac{1}{T^2}\right)$	$O\left(\frac{1}{T^2}\right)$
HB*, fixed momentum	$O\left(\frac{T}{n}\right)$	$O\left(\frac{1}{T}\right)$	$O\left(\frac{1}{T}\right)$
SGD, $\eta = 1/L$	$O\left(\frac{T}{n}\right)$	$O\left(\frac{1}{T} + C\right)$	$O\left(\frac{1}{T}\right)$
SGD, $\eta = O\left(T^{-m}\right)$	$O\left(\frac{T^{1-m}}{n}\right)$	$O\left(\frac{1}{T^{1-m}}\right)$	$O\left(\frac{1}{T^{1-m}}\right)$
SGLD, $\eta = O\left(T^{-1}\right)$	$O\left(\frac{T^{1/4}}{n}\right)$	–	$O\left(\frac{1}{T^{1/4}}\right)$

Table 5.1. Uniform stability and convergence lower bound under convex smooth setting. *Stability results for NAG and HB are only proved for quadratic loss and so the convergence lower bound in the same row is conjectured. C is some universal constant, meaning that SGD with constant step-size does not converge to optimum. We are not aware of the convergence rate upper bound of SGLD.

order $O(1/T)$. However, it is known that fixed-step-size stochastic gradient descent can not converge arbitrarily small error at the rate $O(1/T)$ [48]. The best rate of convergence to minimize a smooth non-strongly convex function with noisy gradients is of order $O(T^{-\frac{1}{2}})$ [146]. Therefore, in the case of fixed step-size SGD, the convergence lower bound we provide is valid but loose. The fixed step-size SGD is a stable algorithm but is not a convergent algorithm.

On the other hand, it is shown in the same work [146] that $O(T^{-\frac{1}{2}})$ convergence rate is achieved by stochastic gradient descent with decreasing step-size of order $O(T^{-\frac{1}{2}})$. Using our stability argument, we provide insights why the stochastic gradient descent with decreasing step-size is not converging too fast. It has also been shown by Hardt et al. [75] that stochastic gradient descent with decreasing step-size of order $O(T^{-\frac{1}{2}})$ has $O(\sqrt{T}/n)$ uniform stability. Applying Corollary 5, we conclude that when this decreasing step-size is used, gradient descent can not converge as fast as $O(T^{-1})$.

Similar arguments can be used to explain the conjecture by Moulines and Bach [139] on the optimal convergence rates for stochastic gradient descent of $O(T^{-m})$ step-size. It is shown in [139] that, for $m \in (2/3, 1)$, the convergence rate of stochastic gradient descent for the convex L-smooth case is upper bounded by $O(T^{m-1})$. It is shown by Hardt et al. [75] that stochastic gradient descent of $O(T^{-m})$ step-size has $O(T^{1-m}/n)$ uniform stability in this set-up. Applying Corollary 5, we provide a proof of this conjecture, confirming the optimality of this convergence rate upper bound.

Nesterov accelerated gradient descent

According to Theorem 10, the Nesterov accelerated gradient descent with fixed step-size has $\frac{4\eta(RL)^2T^2}{n}$-uniform stability for quadratic loss functions. Under the conjecture that the same stability holds for convex smooth loss functions, according to Corollary 5, we could obtain that its convergence rate is lower bounded by

$$\mathcal{E}_{\text{opt}}^{\text{NAG}}(T, \mathcal{L}_{\text{c}}) \geq \frac{R^2}{4C_2\eta T^2}. \tag{5.9}$$

This is compatible with its convergence rate upper bound provided in [148]. For f convex and L-smooth function, Nesterov accelerated gradient method with step-size $\eta \leq \frac{1}{L}$ satisfies

$$f(\theta_T) - f(\theta^*) \leq \frac{2\|\theta_1 - \theta^*\|^2}{\eta T^2}.$$

We can compare our stability based lower bounds to classical ways of getting complexity lower bound using the classical first-order oracle of complexity [147, 149]. The classical oracle model based lower bound provides $O(1/T^2)$ lower bound for all first order optimization methods that falls into the following black-box framework. It assumes that the optimization methods takes initialization $\theta_1 = 0$ and at iteration t, θ_t is in the linear span of all previous gradients. Whereas our results show that all optimization methods with order $O(T^2/n)$ uniform stability in the smooth non-strongly convex setting would have convergence rate lower bounded by $O(1/T^2)$. The two lower bounds have similar form, but apply under different scenarios. One remarkable property of our result is that it does not depend on how exactly the algorithm is initialized.

Heavy ball method with fixed step-size

According to Theorem 11, heavy ball method with fixed step-size $\eta \in \left(0, \frac{(1-\gamma)}{L}\right)$ and fixed momentum parameter $\gamma \in [0, 1)$ has

$$\frac{4\eta M^2 T}{(1-\sqrt{\gamma})n}.$$

uniform stability for quadratic loss functions. Under the conjecture that the same stability holds for convex smooth loss functions, applying Corollary 5, we obtain that its convergence rate is lower bounded by $O(1/T)$. First, this lower bound matches the convergence rate upper bound proved in [69]. Second, unlike Nesterov accelerated gradient descent, even though multiple steps of gradients are used, heavy ball method with fixed step-size is not able to achieve the optimal convergence rate $O(1/T^2)$. Another viewpoint on this result is that the smart choice of weighting coefficients in NAG is necessary to its optimal convergence guarantee.

Stochastic gradient Langevin dynamics (SGLD)

According to [138], stochastic gradient Langevin dynamics with temperature τ and decreasing step-size $O(1/T)$, when used for convex optimization, has

$$O(\frac{M^2 \left(\kappa\eta_0\right)^{1/2} T^{1/4}}{n})$$

uniform-stability. Applying Corollary 5, we conclude that its convergence rate is lower bounded by $O(1/T^{1/4})$. While the additional noise added in SGLD might be helpful for certain non-convex optimization settings in escaping local minima as stated in [138], SGLD has a slower worst-case convergence than the GD or SGD based on our stability argument.

5.4.3 Stability in the strongly convex smooth setting

In this subsection, we establish uniform stability for gradient descent, Nesterov accelerated gradient method in the strongly convex smooth setting. In the strongly convex smooth setting, the loss function $l(\cdot, z)$ is m strongly-convex, L-smooth for every $z \in \mathcal{Z}$.

Gradient descent (GD)

The gradient descent method in the strongly convex setting has exactly the same updates as before, given a strongly convex smooth objective F, for $t \geq 0$,

$$\theta_{t+1} = \theta_t - \eta \nabla F(\theta_t),$$

where $\eta \leq 1/L$ is the step-size. While the algorithm stays the same, the strongly convex property of the loss function allows the algorithm to have a better stability.

Theorem 12. *Given a data distribution P, under the assumption that $l(\cdot, z)$ is m-strongly convex, L-smooth and M-Lipschitz for every $z \in \mathcal{Z}$, the full gradient method with constant step-size $\eta \leq \frac{1}{L}$, which outputs $\hat{\theta}_T$ at iteration $T \geq 1$, has uniform stability*

$$\mathcal{E}_{stab}^{GD,\ strongly\ convex}(\hat{\theta}_T, l, P, n) \leq \frac{4M^2}{mn}\left(1 - \left(1 - \frac{\eta L}{1+\kappa}\right)^T\right). \tag{5.10}$$

The proof of this theorem is provided in Appendix C.2.1.

Stochastic gradient descent (SGD) with fixed step-size

The stochastic gradient descent in the strongly convex setting has the exactly same updates as before. It starts at some initial point $\theta_0 \in \Omega$, and iterates with the following recursion with i chosen from the set $\{1, ..., n\}$ uniformly at random,

$$\theta_{t+1} = \theta_t - \eta \nabla f_i(\theta_t).$$

The stability of SGD under strongly convex setting has been first discussed in [75]. According to Theorem 3.10 in Hardt et al. [75], the stability of SGD under strongly convex setting is upper bounded by

$$\mathcal{E}_{\text{stab}}^{\text{SGD, fixed, strongly convex}}(\hat{\theta}_T, l, P, n) \leq \frac{2M^2}{mn}\left(1-(1-\eta m/2)^T\right) \tag{5.11}$$

at iteration $T \geq 1$, for any m-strongly convex, M-Lipschitz and L-smooth loss function l.

Nesterov accelerated gradient descent (NAG)

Unlike in the convex smooth setting, Nesterov's accelerated gradient descent can take fixed momentum parameter in the strongly convex smooth setting.

$$\begin{aligned}\theta_{t+1} &= w_t - \eta \nabla F(w_t)\\ w_{t+1} &= \left(1+\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)\theta_{t+1} - \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\theta_t,\end{aligned}$$

where $\eta \leq \frac{1}{L}$ is the step-size, $\kappa = L/m$.

We prove its uniform stability for m strongly-convex, L-smooth for quadratic loss function.

Theorem 13. *Given a data distribution P, under the assumption that $l(\cdot, z)$ is m-strongly convex, L-smooth and M-Lipschitz for every $z \in \mathcal{Z}$, Nesterov accelerated gradient descent method described above, which outputs $\hat{\theta}_T$ at iteration $T \geq 1$, has uniform stability*

$$\mathcal{E}_{stab}^{NAG,\ strongly\ convex}(\hat{\theta}_T, l, P, n) \leq \frac{4M^2}{mn}\left(1-\left(1-\frac{1}{\sqrt{\kappa}}\right)^T\right). \tag{5.12}$$

The proof of this theorem is provided in Appendix C.2.2.

5.4.4 Consequences for the convergence lower bound in the strongly convex setting

In this subsection, we obtain convergence lower bound for GD and NAG in the m-strongly convex L-smooth setting via Theorem 8. In Table 5.2, we summarize all the uniform stability results and the corresponding convergence lower bounds under strongly convex smooth setting. While exact constants are provided in the main text, we only show the dependency on iteration number T and sample size n in the table.

Method	Uniform stab.	Conv. upper (known)	Convergence lower (ours)
GD	$O\left(\frac{1}{n}\left(1-e^{-O(T/\kappa)}\right)\right)$	$e^{-O(T/\kappa)}$	$e^{-O(T/\kappa)}-C$
NAG*	$O\left(\frac{1}{n}\left(1-e^{-O\left(T/\sqrt{\kappa}\right)}\right)\right)$	$e^{-O\left(T/\sqrt{\kappa}\right)}$	$e^{-O\left(T/\sqrt{\kappa}\right)}-C$
SGD	$O\left(\frac{1}{n}\left(1-e^{-O(T/\kappa)}\right)\right)$	$e^{-O(T/\kappa)}+C$	$e^{-O(T/\kappa)}-C$

Table 5.2. Uniform stability and convergence lower bound under strongly convex setting. *Stability results for NAG are only proved for quadratic loss and so the convergence lower bound in the same row is conjectured. C is some universal constant, meaning that SGD with constant step-size does not converge to optimum and our convergence lower bound has an undesirable offset in this setting.

Gradient descent

According to Theorem 12, gradient descent with fixed step-size η in the strongly convex smooth setting has

$$\frac{4\left(RL\right)^2}{mn}\left(1-\left(1-\frac{\eta L}{1+\kappa}\right)^T\right)$$

uniform stability. We apply Theorem 8 to obtain a lower bound on the convergence of GD for strongly convex smooth functions.

$$\mathcal{E}_{\text{opt}}^{\text{GD}}(T,\mathcal{L}_{\text{sc}}) \geq \frac{LR^2}{C_3 n} - \frac{4\left(RL\right)^2}{mn} + \frac{4\left(RL\right)^2}{mn}\left(1-\frac{\eta L}{1+\kappa}\right)^T. \tag{5.13}$$

If the leading constants $\frac{LR^2}{C_3}$ and $\frac{4(RL)^2}{m}$ match, we could directly obtain a lower bound on its convergence of order $e^{-O(T/(1+\kappa))}$ as we expect. Unfortunately, due to our proof of the empirical risk minimization lower bound, a couple factors of constants are lost. Thus directly applying the stability bound makes it impossible to match the leading constants. We always have

$$\frac{4\left(RL\right)^2}{mn} \geq \frac{LR^2}{C_3 n}.$$

Therefore, our trade-off result only gives convergence lower bound of GD with an offset of $\frac{LR^2}{C_3 n} - \frac{4(RL)^2}{mn}$ as stated in Equation (5.13).

Remark that a similar lower bound can be obtained for stochastic gradient descent using exactly the same argument for GD.

Nesterov accelerated gradient descent

According to Theorem 13, Nesterov accelerated gradient descent with fixed step-size η in the strongly convex smooth setting has

$$\frac{4M^2}{mn}\left(1-\left(1-\frac{1}{\sqrt{\kappa}}\right)^T\right)$$

uniform stability for quadratic loss function. Since the construction of the minimax lower bound in Theorem 8 is based on quadratic loss functions, applying Theorem 8 by restricting to quadratic loss functions, we obtain an expected convergence lower bound of order $e^{-O\left(T/\sqrt{\kappa}\right)}$ with an offset,

$$\mathcal{E}_{\text{opt}}^{\text{NAG}}(T,\mathcal{L}_{\text{sc}}) \geq \frac{LR^2}{C_3 n} - \frac{4\left(RL\right)^2}{mn} + \frac{4\left(RL\right)^2}{mn}\left(1-\frac{1}{\sqrt{\kappa}}\right)^T. \tag{5.14}$$

5.5 Proof of Main Results

In this section, we prove the main trade-off results.

5.5.1 Proof of Theorem 7

Using Equation (5.2), Theorem 7 directly follows from the well-known statistical lower bound for empirical risk estimation with adaptation to convex smooth loss functions. For completeness, we restate this lower bound and provide the proof below.

Lemma 16. *For any fixed sample size n, there exists a universal constant $C_1 > 0$ and L-smooth convex loss function l defined on $\mathcal{Z}\times\Omega$, with $R = |\Omega|$, such that*

$$\inf_{\hat{\theta}}\sup_{P\in\mathcal{P}}\mathbb{E}_S\left[\delta R(\hat{\theta})\right] \geq \frac{R^2 L}{C_1\sqrt{n}}.$$

Proof of Lemma 16 The main idea to prove this lemma is to formulate the excess risk minimization problem as binary hypothesis testing problem and then apply Le Cam's method for lower bound.

For any fixed sample size n, define domain $\mathcal{Z}$ be $\{-1,1\}$ and two probability distributions P_1 and P_2 satisfying the following two properties,

$$P_1\left(Z=-1\right) = P_2\left(Z=1\right) = \frac{1}{2}+\frac{1}{\sqrt{24n}},$$
$$P_1\left(Z=1\right) = P_2\left(Z=-1\right) = \frac{1}{2}-\frac{1}{\sqrt{24n}}.$$

We define P_1^n to be the joint distribution where $Z_1, \ldots, Z_n$ are independent samples from P_1, and we defin P_2 accordingly.

Let $\theta_1^* \in \Omega$ with all other coordinates zero but the first coordinate equals to $-\delta$, and $\theta_2^* \in \Omega$ with all other coordinates zero but the first coordinate equals to δ, with $0 < \delta \leq r$. δ and r are a constants to be determined later. Let $\theta[1]$ be the first coordinate of θ and let $\Phi(r)$ be the parameter such that

$$\forall v \in \{1,2\}, |\theta[1] - \theta_v^*[1]| \geq r \Rightarrow \mathbb{E}_{Z\sim P_v}\left[\delta R(\theta')\right] \geq \Phi(r).$$

The exact form of $\Phi(r)$ will be determined after we define the loss function l. We have

$$\inf_{\hat{\theta}\in\Omega} \max_{v\in\{1,2\}} \mathbb{E}_{P_v}\left[\delta R(\hat{\theta})\right] \geq \Phi(r) \cdot \inf_{\hat{\theta}\in\Omega} \max_{v\in\{1,2\}} P_v^n\left(\left|\hat{\theta}[1] - \theta_v^*[1]\right| \geq r\right). \tag{5.15}$$

Le Cam's method reduce this estimation problem to binary hypothesis testing problem, then we have

$$\inf_{\hat{\theta}\in\Omega} \max_{v\in\{1,2\}} P_v^n\left(\left|\hat{\theta}(Z_v^n)[1] - \theta_v^*[1]\right| \geq r\right) \geq \inf_{\Psi} \max_{v\in\{1,2\}} P_v^n\left(\Psi(Z_v^n) \neq v\right),$$

where the infimum ranges over all testing functions $\Psi : \mathcal{Z}^n \to \{1,2\}$.

We have for any $\Psi : \mathcal{Z}^n \to \{1,2\}$ that the probability of error is

$$\max_{v\in\{1,2\}} P_v^n\left(\Psi(Z_v^n) \neq v\right) = \frac{1}{2}P_1^n\left(\Psi\left(Z_1^n\right) \neq 1\right) + \frac{1}{2}P_2^n\left(\Psi\left(Z_2^n\right) \neq 2\right)$$

A standard result of [104] gives the exact expression of the minimal possible error in the above hypothesis test. We have

$$\inf_{\Psi}\left\{P_1^n\left(\Psi\left(Z_1^n\right) \neq 1\right) + P_2^n\left(\Psi\left(Z_2^n\right) \neq 2\right)\right\} = 1 - \left\|P_1^n - P_2^n\right\|_{\mathrm{TV}},$$

where $\|\cdot\|_{\mathrm{TV}}$ denotes the total variation distance. Using Pinsker's inequality, we have

$$\begin{aligned}
\left\|P_1^n - P_2^n\right\|_{\mathrm{TV}}^2 &\leq 2\mathrm{KL}\left(P_1^n || P_2^n\right) \\
&= \frac{n}{2}\mathrm{KL}\left(P_1 || P_2\right) \\
&\overset{(i)}{=} \frac{n}{2} \cdot \frac{1}{\sqrt{6n}} \log \frac{1 + \frac{1}{\sqrt{6n}}}{1 - \frac{1}{\sqrt{6n}}} \\
&\overset{(ii)}{\leq} \frac{n}{2} \cdot \frac{3}{6n} \\
&= \frac{1}{4}
\end{aligned}$$

.

Equality (i) uses the KL divergence formula between two Bernoulli distributions. Inequality (ii) uses the inequality $\delta \log \frac{1+\delta}{1-\delta} \leq 3\delta^2$ for $\delta \in \left[0, \frac{1}{2}\right]$. Thus, we show that any

test Ψ will mistake one of the probability distribution for the other with probability at least $\frac{1}{4}$.

$$\inf_{\Psi} \max_{v\in\{1,2\}} P_v^n \left(\Psi(Z_v^n) \neq v\right) \geq \frac{1}{4}.$$

It remains to design a L-smooth convex loss function l and determine the exact form of Φ. Without loss of generality, we can assume that Ω is center around 0. We define the loss function $l(\theta; z)$ to be

$$l(\theta;-1) = \begin{cases} \frac{L}{2}\left(\theta[1]+r\right)^2 & \text{for } |\theta[1]+r| \leq \frac{r}{2} \\ \frac{Lr}{4}|\theta[1]+r| & \text{otherwise,} \end{cases}$$

$$l(\theta;1) = \begin{cases} \frac{L}{2}\left(\theta[1]-r\right)^2 & \text{for } |\theta[1]-r| \leq \frac{r}{2} \\ \frac{Lr}{4}|\theta[1]-r| & \text{otherwise.} \end{cases}$$

It is easy to verify that the loss function is convex and L-smooth for each z. Then

$$\mathbb{E}_{Z\sim P_1} l\left(\theta; Z\right) = \left(\frac{1}{2} + \frac{1}{\sqrt{24n}}\right) l(\theta;-1) + \left(\frac{1}{2} - \frac{1}{\sqrt{24n}}\right) l(\theta;1).$$

The function $\mathbb{E}_{Z\sim P_1} l\left(\theta; Z\right)$ is differentiable along the first coordinate. Its derivative is nondecreasing and vanishes on the interval $\left[-r, -\frac{r}{2}\right]$. Thus the minimizer $\theta_1^*[1]$ falls into the interval $\left[-r, -\frac{r}{2}\right]$.

For $\theta' \in \Omega$ such that $|\theta'[1] - \theta_1^*[1]| \geq r$, using the derivative of $\mathbb{E}_{Z\sim P_1} l\left(\theta; Z\right)$, we have

$$\mathbb{E}_{Z\sim P_1}\left[\delta R(\theta')\right] \geq \min\left\{\mathbb{E}_{Z\sim P_1}\left[\delta R(0)\right], \mathbb{E}_{Z\sim P_1}\left[\delta R(\theta_{1,\text{left}})\right]\right\}$$

where $\theta_{1,\text{left}}$ is zero everywhere but $-\frac{3r}{2}$ on the first coordinate. Then

$$\mathbb{E}_{Z\sim P_1}\left[\delta R(\theta')\right] \geq \frac{Lr^2}{\sqrt{96n}},$$

and the same holds for P_2. Plugging $\Phi(r) = \frac{Lr^2}{\sqrt{96n}}$ into Equation (5.15), we can conclude that

$$\inf_{\hat{\theta}\in\Omega} \max_{v\in\{1,2\}} \mathbb{E}_{P_v}\left[\delta R(\hat{\theta})\right] \geq \frac{Lr^2}{\sqrt{96n}} \cdot \frac{1}{4} \geq \frac{Lr^2}{16\sqrt{6n}}.$$

We remark that we can take r as large as $\frac{R}{2}$. Thus we conclude that

$$\inf_{\hat{\theta}\in\Omega} \max_{v\in\{1,2\}} \mathbb{E}_{P_v}\left[\delta R(\hat{\theta})\right] \geq \frac{R^2 L}{256\sqrt{6n}}.$$

5.5.2 Proof of Corollary 5

Applying Theorem 7, for any sample size n and T, we have

$$\frac{s(T)}{n} + \mathcal{E}_{\text{optimization}}(T, n) \geq \frac{R^2 L}{C_1 \sqrt{n}}.$$

As we only consider optimization method designed for any convex problems, $\mathcal{E}_{\text{optimization}}$ is independent of the sample size n. This result is valid for any sample size n. We can take n such that the following quadratic function

$$Q(\frac{1}{\sqrt{n}}) = \frac{R^2 L}{C_1 \sqrt{n}} - \frac{s(T)}{n},$$

is maximized to obtain the best lower bound.

Completing the square, we have

$$Q(n) = -s(T) \left(\frac{1}{\sqrt{n}} - \frac{R^2 L}{2 C_1 s(T)} \right)^2 + \frac{R^4 L^2}{4 C_1^2 s(T)}.$$

$\frac{2C_1 s(T)}{R^2 L}$ would be the best choice of $\sqrt{n}$, but we have to ensure that n is an integer. Since $s(T)$ is divergent function of T, there exists $T_0 \geq 1$, such that for $T \geq T_0$, we can always find integer n satisfying

$$\frac{4C_1 s(T)}{3R^2 L} \leq \sqrt{n} \leq \frac{4C_1 s(T)}{R^2 L}.$$

Plugging n, we conclude that there exists universal constant C_2, and a convex function such that for $T \geq T_0$,

$$\mathcal{E}_{\text{optimization}}(T, n) \geq \frac{R^4 L^2}{C_2 s(T)}.$$

5.5.3 Proof of Theorem 8

We prove the statistical lower bound for empirical risk estimation in the strongly convex case via similar techniques used in the proof of Lemma 16. Le Cam's argument for reducing an estimation problem to binary hypothesis testing problem is still valid. All we do is to define a m-strongly convex L-smooth loss function l and find the corresponding $\Phi(r)$. We define the loss function $l(\theta; z)$ to be

$$l(\theta; -1) = \frac{L}{2} \left(\theta[1] + r \right)^2,$$
$$l(\theta; 1) = \frac{L}{2} \left(\theta[1] - r \right)^2.$$

l is quadratic, so it is m-strongly convex and L smooth for each z. Then

$$\begin{aligned}\mathbb{E}_{Z\sim P_1} l(\theta; Z) &= \left(\frac{1}{2} + \frac{1}{\sqrt{24n}}\right) l(\theta; -1) + \left(\frac{1}{2} - \frac{1}{\sqrt{24n}}\right) l(\theta; 1) \\ &= \frac{L}{2}\left(\theta[1]^2 + \frac{2}{\sqrt{6n}}\theta[1]r + r^2\right) \\ &= \frac{L}{2}\left(\theta[1] + \frac{r}{\sqrt{6n}}\right)^2 + \frac{L}{2}\left(r^2 - \frac{r^2}{6n}\right).\end{aligned}$$

The minimizer θ_1^* has the first coordinate equals to $-\frac{r}{\sqrt{6n}}$. And the minimum is $\frac{L}{2}\left(r^2 - \frac{r^2}{6n}\right)$.

For $\theta' \in \Omega$ such that $|\theta'[1] - \theta_1^*[1]| \geq r$, we have

$$\mathbb{E}_{Z\sim P_1} l(\theta'; Z) \geq \frac{Lr^2}{2}.$$

Thus, we have

$$\mathbb{E}_{Z\sim P_1}\left[\delta R(\theta')\right] \geq \frac{Lr^2}{12n}$$

The same lower bound holds for P_2. Plugging $\Phi(r) = \frac{Lr^2}{12n}$ into Equation (5.15), we can conclude that

$$\inf_{\hat{\theta}\in\Omega} \max_{v\in\{1,2\}} \mathbb{E}_{P_v}\left[\delta R(\hat{\theta})\right] \geq \frac{Lr^2}{12n}\cdot\frac{1}{4} \geq \frac{Lr^2}{48n}.$$

We remark that we can take r as large as $\frac{R}{2}$. Thus we conclude that

$$\inf_{\hat{\theta}\in\Omega} \max_{v\in\{1,2\}} \mathbb{E}_{P_v}\left[\delta R(\hat{\theta})\right] \geq \frac{R^2 L}{192n}.$$

5.6 Simulations

In this section, we first show via simulation results of a simple logistic regression applied on breast-cancer-wisconsin dataset that the stability bounds established in this chapter have the right scaling on the iteration number T. Second, we illustrate via a logistic regression problem that the stability bound characterize better the generalization error than simple uniform convergence bound at least for the first iterations of GD and NAG.

5.6.1 Algorithmic Stability Rate Scaling

We evaluate our stability bounds for all gradient methods mentioned on logistic regression with the binary classification datasets breast-cancer-wisconsin [198]. This dataset

has sample size $n = 699$ and dimension $d = 10$. The problem of logistic regression is formulated as follows.

Given a set of i.i.d. samples $\{(X_i, Y_i)\}_{i=1}^n$, with $X_i \in \mathbb{R}^d$ and $Y_i \in \{0, 1\}$, we want to estimate the parameter θ which characterizes the conditional distribution of Y_1 given X_1:

$$\mathbb{P}(Y_i = 1 | X_i; \theta) = r(\theta, X_i) = \frac{e^{\theta^\top X_i}}{1 + e^{\theta^\top X_i}}.$$

Let $Y = (Y_1, \ldots, Y_n)^\top \in \{0, 1\}^n$ and X be the $n \times d$ matrix with X_i as i^{th}-row. The log-likelihood function we optimize over is as follows,

$$f(\theta) = \frac{1}{n} \left(-Y^\top X\theta + \sum_{i=1}^n \log \left(1 + e^{\theta^\top X_i} \right) \right). \tag{5.16}$$

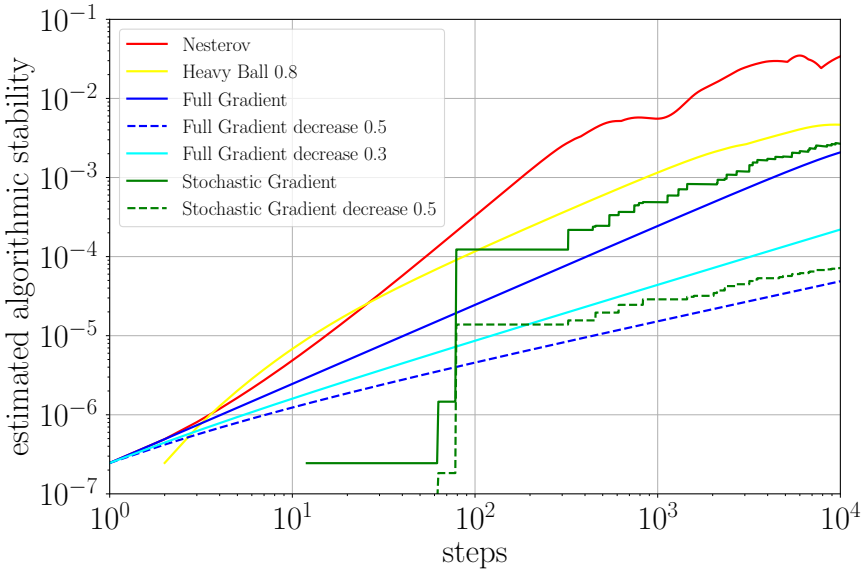


Figure 5.1. Estimated algorithmic stability of various gradient methods mentioned with independent 50 runs. The estimated uniform stabilities of full gradient method, stochastic gradient method and heavy ball method with fixed step-size all have slope 1 in log-log plot, while Nesterov accelerated gradient method has slope 2. Methods with decreasing step-size have a slope smaller than 1.

It can be shown that this objective has the Lipschitz constant M equal to 1 and the smoothness parameter L equal to $1/4$ when the covariate matrix X is normalized to have its maximum eigenvalue equal to 1. When there is no regularization, each loss function f_i is not strongly convex $\mu = 0$. In all of our experiments we set constant step-size $\eta = 0.1$. To construct samples that differ only on one data point, we first fix a sample S with size 500 from dataset, then construct a perturbed sample S' by changing

one data point in S and finally run our optimization algorithm to compute and plot the model difference $\|\theta_t - \theta'_t\|_2$. The norm difference $\|\theta_t - \theta'_t\|_2$ constitute an estimate for the uniform stability up to constants independent of T and n. Finally, the perturbation on the sample is repeated 50 times. Figure 5.1 shows the estimated uniform stability, averaged over 50 independent repeats, for all gradient methods methods, Nesterov accelerated gradient, heavy ball method with fixed momentum ($\gamma = 0.8$), full gradient method with fixed step-size, full gradient method with decreasing step-size T^{-m} ($m = 0.5, 0.3$), stochastic gradient method with fixed step-size and stochastic gradient method with decreasing step-size T^{-m} ($m = 0.5$). We observe that the estimated uniform stabilities of full gradient method, stochastic gradient method and heavy ball method with fixed step-size all have slope 1 in log-log plot, while Nesterov accelerated gradient method has slope 2. As expected, methods with decreasing step-size have a slope smaller than 1. Even though the stability bounds of NAG and HB are only established for quadratic loss, the estimated stability in the simulation makes us conjecture that the stability bounds of NAG and HB still hold in the general convex smooth setting.

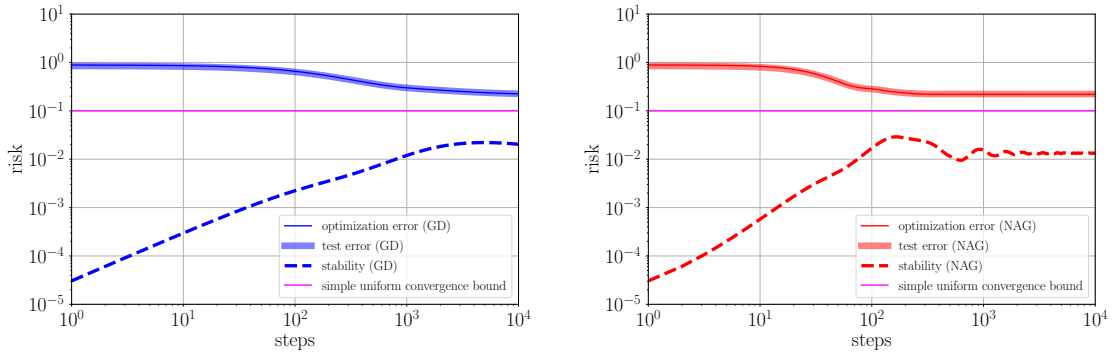
5.6.2 Algorithmic stability vs simple uniform convergence bounds

The goal of this simulation is to show that algorithmic stability characterize the generalization error better than the simple uniform convergence bounds, which can not easily take into account of the growth of the function space for iterative algorithms. For d-dimensional estimation problem, simple uniform convergence bound would give an generalization error bound of order $O\left(\sqrt{d/n}\right)$. The exact constant in the uniform convergence bound depends on the function space and is hard to characterize for iterative algorithms. We think that more refined uniform convergence bound via Rademacher complexity [11] might be possible, but we are not aware of such results for general iterative algorithms. In this section, we show via simulations that the simple uniform convergence bound of order $O\left(\sqrt{d/n}\right)$ is less precise than the stability in characterizing generalization error. More precisely, we can see that when the dimension d and the number of samples n are large and iteration T is small

$$\sqrt{\frac{d}{n}} \gg \frac{s(T)}{n},$$

where $s(T)/n$ is the stability bound for GD or NAG. We show in the next two experiments that this comparison is valid and the stability bound is more relevant in large scale problems.

In the both experiments, we fix the true parameter $\theta^* = (1, \ldots, 1)^\top$ and we random draw n i.i.d. samples (X_i, Y_i) according to the following data generation process. Each row of X is drawn from a standard d-dimensional normal distribution, and then X is renormalized to have row norm 1. Each label Y_i, give $X_i = x$, is drawn from a Bernoulli



(a) Exp 1. Gradient Descent (b) Exp 1. Nesterov Accelerated Gradient

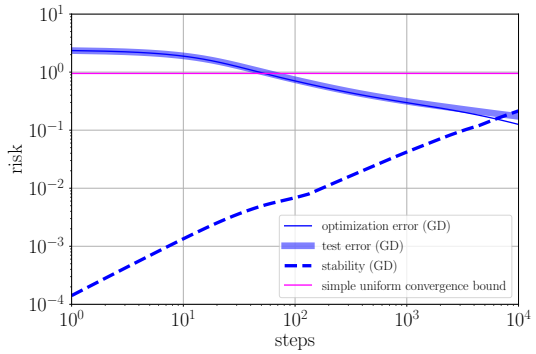
Figure 5.2. Algorithmic stability vs simple uniform convergence bound in the first experiment, $d = 20, n = 2000$. For both GD and NAG, the optimization error plot aligns with the test error plot, indicating that optimization error dominates in the risk decomposition. Whether using stability or simple uniform convergence bound to characterize generalization error is not important.

distribution with parameter $r(\theta^*, x)$. We use both the gradient descent and Nesterov accelerated gradient to optimize the empirical log-likelihood objective in Equation (5.16). We estimate the stability using its definition in Equation (5.1) by varying different z from holdout data set. In first experiment, we set $d = 20, n = 2000$. Figure 5.2 shows that both the simple uniform convergence bound and estimated stability bound are small compared to optimization error. In this setting, driving optimization error to zero is more important for reducing the test error, as shown in thick red color. We can still observe that the scalings of the estimated stability bound for GD and NAG are different. Our theoretical stability bound follows the estimated stability bound with the same slope, but without the saturation at the end of iterates.

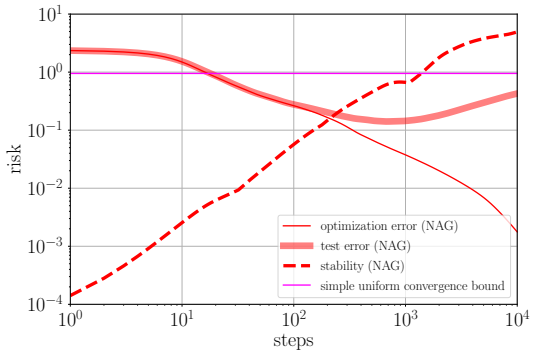
In the second experiment, we set $d = 200, n = 2000$. Figure 5.3 shows that the generalization error accounts for a large portion of the test error. Especially, we observe in Figure 5.3b that the test error of NAG deviates from its training error. Simple uniform convergence bound does not explain the overfitting phenomenon here. The algorithmic stability combined with the training error suggests that early-stopping should be used for NAG in this setting as shown in Figure 5.4.

5.7 Summary

In this chapter, we have shown the trade-off between stability and convergence rate for several algorithms. Here we briefly discuss how our stability bound for optimization could served as an early stopping criteria. Additionally, we consider possible other iterative algorithms such as boosting that could fit into this stability and optimization trade-off framework.

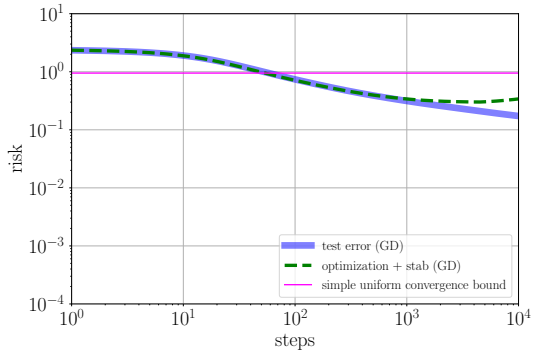


(a) Exp 2. Gradient Descent

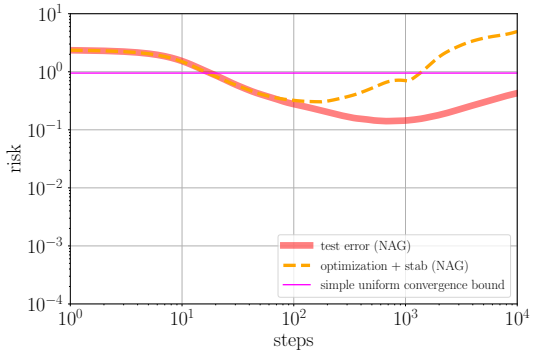


(b) Exp 2. Nesterov Accelerated Gradient

Figure 5.3. Algorithmic stability vs simple uniform convergence bound in the second experiment, $d = 200, n = 2000$. As the test error deviates from the optimization error, the generalization error accounts for a large portion of the test error. Because the simple uniform convergence bound does not depend on the iteration number, it can't explain the overfitting phenomenon especially for NAG.



(a) Exp 2. Gradient Descent



(b) Exp 2. Nesterov Accelerated Gradient

Figure 5.4. Stability + optimization error in the second experiment, $d = 200, n = 2000$. Stability + optimization error shown in dashed line aligns well with the test error curve.

Minimizing empirical risk is often computationally expensive in large scale learning problems. As it has been pointed out in [19], optimization algorithms do not need to carry out this minimization with great accuracy since the empirical risk is already an approximation to the expected risk. For example, we can stop an iterative optimization algorithm long before its convergence to reduce computational cost. How early we should stop without deteriorating too much the expected risk becomes the main question we ask in large scale learning problems. The expected excess risk decomposition has been the main theoretical guideline for this kind of early-stopping criteria. Even though in this reasoning we are studying upper bounds of generalization and optimization errors, it is often accepted that these upper bounds give a realistic idea of the actual

convergence rates [192, 20, 10, 19].

We would like to stop our optimization algorithm as far as it reaches an optimization error close to its generalization error. However, the uniform convergence bounds are often too pessimistic about the size of the space to search over. Instead, we use our stability based generalization bound as an estimate of the generalization error. Formally, we would choose iteration T such that

$$\mathcal{E}_{\text{stab}}(T, n) \approx \mathcal{E}_{\text{opt}}(T).$$

As an example, our stability based generalization bound for fixed-step-size full gradient method in the smooth non-strongly convex setting is $\frac{2\eta M^2 T}{n}$. The first remarkable point is that this generalization error bound is dimension-free. Because it is often hard to access accurate estimates for the uniform convergence bounds based generalization error, it might be advantageous to acquire a theoretical early-stopping criterion via our stability bounds. For the full gradient method trained model, as long as the Lipschitz constant M and smoothness constant L can be estimated accurately, we are able to give an early stopping criterion such as $T \approx \sqrt{\frac{n}{\eta^2 M^2 R^2}}$, given the estimate of R is accurate.

Boosting is one of the most successful and practical iterative optimization methods. Unlike gradient method which iterates over parameters, boosting starts with a sensible estimator or classifier, the learner, and seeks its improvements iteratively on the function space. The bias-variance trade-off of L2 boosting discussed in Buhlmann et al. [26] shares similar behaviors as the trade-off we discussed in Equation (5.2). It would be interesting to characterize the stability of boosting algorithms with various kinds of weaker learners and derive precise trade-off results as we did for gradient methods.

Part IV

Applications in Neuroscience

Chapter 6

DeepTune: data-driven visualization of V4 tuning

Understanding how primates process visual information and recognize objects in an image is a major problem in neuroscience. Along the visual pathway, the mid-tier cortical area V4 is of particular interest. Despite its importance in the hierarchical organization of visual processing, its function remains elusive. Deep neural network models have recently been shown to be effective in predicting single neuron responses in primate visual cortex areas V4. Despite their high predictive accuracy, these models are generally difficult to interpret. This limits their applicability in characterizing V4 neuron function. Here, we propose the DeepTune framework as a data-driven way via optimization and sampling to elicit interpretations of deep neural network-based models of single neurons in area V4. Using a dataset of recordings of 71 V4 neurons stimulated with thousands of static natural images, we build an ensemble of 18 neural network-based models per neuron that accurately predict its response given a stimulus image. To interpret and visualize these models, we use a stability criterion to form optimal stimuli (DeepTune images) by pooling the 18 models together. These DeepTune images not only confirm previous findings on the presence of diverse shape and texture tuning in area V4, but also provide rich, concrete and naturalistic characterization of receptive fields of individual V4 neurons. The population analysis of DeepTune images for 71 neurons reveals how different types of curvature tuning are distributed in V4.

6.1 Introduction

Understanding the function of primate visual pathways is a major challenge in computational neuroscience. Along the ventral visual pathway, cortical area V4 is of particular interest. It is a large retinotopically-organized area located intermediate between the early primate visual cortex areas such as V1 and V2 and high-level areas in the inferior temporal (IT) lobe. V4 is believed to be crucial for visual object recognition and visual attention, but its functional role remains mysterious. Computational studies of

primary visual cortex have produced powerful quantitative models of V1 [29]. Contrastingly, area V4 is more difficult to model computationally than V1. This is mainly due to its highly nonlinear response [189] and diverse tuning properties [169].

To understand the tuning properties of V4 neurons, one dominant traditional approach is to use handcrafted and limited synthetic stimuli (e.g. [64, 153]) to probe V4 neurons. For example, by comparing V4 neuron responses to Cartesian gratings with those to polar and hyperbolic (non-Cartesian) gratings, Gallant et al. [64, 65] found that V4 neurons are most selective for non-Cartesian gratings containing multiple orientations. Through a parameterized set of contour stimuli varying in angularity, curvature, and orientation, Pasupathy and Connor [154, 153] discovered that V4 neurons are selective to curved contour features. While such studies have successfully quantified V4 neuron responses to synthetic shapes, the tuning properties of most V4 neurons cannot be fully explored through these limited sets of stimuli [169].

An alternative approach to designing synthetic stimuli is using a large collection of natural images directly as stimuli. This approach reduces the difficulty in stimuli design, but creates a huge challenge in modeling. Specifically, it has been found that previously proposed simple and shallow computational models of V4 neurons perform poorly on natural images [46, 143, 169]. For instance, David et al. [46] introduced the spectral receptive field (SRF) model to account for second order nonlinear response properties. The SRF model enhances our understanding of V4 orientation tuning properties, but its average prediction performance for the V4 neurons studied is far from satisfying [169]. More recently, advances in deep convolutional neural networks (CNNs) with multiple layers of linear and non-linear operations have led to more accurate predictive models for neurons in V4 and IT [204, 28, 203]. While this deep, convolutional and non-linear architecture is the key to the high predictive performance, it also makes the models difficult to interpret. This limits their usefulness in advancing neuroscience. A natural question arises: can we use these complex and accurate models to infer tuning properties of V4 neurons?

In this chapter, we propose the DeepTune framework as a tool to visualize and interpret predictive models of single neurons. In order to make the interpretations be less dependent on arbitrary neural network architecture choices, we build an ensemble of 18 CNN-based models per neuron instead of a single model. The models vary in architecture, but all have comparable high and state-of-the art prediction accuracies. Each model uses a CNN to extract features from an input image. The CNN is pre-trained to perform object classification on the ImageNet dataset [172]. The extracted features are then used as predictors to train a regularized linear regression model with the neuron firing rate as the response. This approach of applying a pre-trained model to a new prediction task is known as transfer learning [177]. For each neuron, we then generate DeepTune images that are obtained via gradient optimization of the fitted models. Aggregating the DeepTune generation process from 18 models via a stability criterion, we further introduce the consensus DeepTune images for each neuron. We show that interpreting the components of DeepTune images that are consistent across 18 models and the consensus one can help better characterize the tuning property of a

neuron and gain robustness against modeling choices. Finally, we perform population analysis of all DeepTune images from 71 neurons to illustrate the curvature tuning diversity and suppressive tuning in V4.

6.2 Data Collection

Extracellular single-unit recordings are made on 71 well-isolated neurons in area V4 of two awake-behaving male macaques. During recordings, subjects performed a fixation task for liquid reward. This recording data was previously used to study the sparseness of neural codes in the area V4 (see Willmore et al. [197] for additional details). The natural images stimuli consist of a random sample of circular patches of grayscale digital photographs from a commercial digital library (Corel, see SI Data Collection for details). Theses images were sampled from the library uniformly at random without replacement. They were concatenated into long sequences to be presented to the macaques. When presented, all images were centered on the estimated classical receptive field (SRF, see SI Data Collection, for CRF estimation procedure). The image size was adjusted to be two to four times the CRF diameter (Figure 6.1-C). The CRF estimation made sure that the receptive fields were approximately centered.

The long sequences of image stimuli was split to form training and test sets. The training set was used to train our model. That is, build a data-driven computational pipeline to relate the image stimuli and neuron responses. The test set was separate from the training set. It was held out during the model training, to consistently evaluate the prediction accuracy and avoid model overfitting. The training set for each neuron contains 4,000-12,000 distinct images. Images were presented to the macaques at 30Hz. Spike counts are recorded at 1ms resolution and then aggregated to the monitor refresh rate 60Hz. Consequently, each image was shown for two consecutive response measurements. The test set for each neuron consists of 300 distinct images, different to the ones in the training set. The images were presented in the same frequency as for the training set. Additionally, the sequence of test images was repeated: for each neuron, each image in the test set was shown 8-10 times. The resulting spike counts were averaged to provide a more precise estimate of the expected spike count. Moreover, repeats also allowed for estimating the amount of variance in the neuron explainable by the stimulus image (15). Finally, the training set (resp. test set) consists of 8,000-24,000 (resp. 600) stimulus-response pairs.

6.3 CNN-based models are highly predictive of V4 stimulus-response data

We introduce a transfer learning framework (Figure 6.1) to build predictive models in two stages for our V4 stimulus-response data as just described. For a given layer of a pre-trained CNN and for each input stimuli, in the first stage (Figure 6.1-A), we

extract intermediate outputs from that layer of CNN as features. In the second stage (Figure 6.1-B), these features serve as predictors in a regularized linear regression (such as Ridge [78] or LASSO [187]) with time-lagged spike rates as the responses. Specifically, for one stimulus image at time t denoted as $\mathbf{z}_t \in \mathbb{R}^{s\times s}$ ($s = 227$ in the AlexNet CNN model [101]), the given layer of CNN transforms this image into a flattened feature vector $\mathbf{x}_t \in \mathbb{R}^d$ ($d = 256 \times 13 \times 13$ in the AlexNet-Layer2 CNN model). This feature transform is denoted as function $h : \mathbb{R}^{s\times s} \mapsto \mathbb{R}^d$. Since the responses of V4 neurons to a sequence of images are sensitive to the recent history of images shown to the subject, we build the models with multiple time lags. More precisely, we regress y_t against the training image features from last k frames of video prior to and including time t, i.e. $\mathbf{z}_t, ..., \mathbf{z}_{t-k+1}$. The time lag k is set to be 9 (consisting frames at 0, 16.7, ..., 133.6 ms) as in previous studies with similar data recordings (e.g. [46, 196]). Finally, our predictive model for a single neuron response takes the following form

$$\begin{aligned} F : \mathbb{R}^{s\times s\times k} &\to \mathbb{R} \\ (\mathbf{z}_t, ..., \mathbf{z}_{t-k+1}) &\mapsto \sum_{j=0}^{k-1} \boldsymbol{\beta}_{j+1}^T h(\mathbf{z}_{t-j}), \end{aligned}$$

where $(\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_k) \in \mathbb{R}^{d\times k}$ are the regression parameters to be estimated and h is the fixed CNN feature transform. The model parameters are learned by solving the following regularized linear regression problem

$$\begin{aligned} \left(\hat{\boldsymbol{\beta}}_1, \ldots, \hat{\boldsymbol{\beta}}_k\right) = \underset{\boldsymbol{\beta}_1,\ldots,\boldsymbol{\beta}_k}{\arg\min} \frac{1}{2} \sum_{t=k}^{T} \left(y_t - \sum_{j=0}^{k-1} \boldsymbol{\beta}_{j+1}^T h(\mathbf{z}_{t-j}) \right)^2 + \\ \lambda_1 \sum_{j=1}^{k} \left\|\boldsymbol{\beta}_j\right\|_1 + \lambda_2 \sum_{j=1}^{k} \left\|\boldsymbol{\beta}_j\right\|_2^2. \end{aligned}$$

If not specified in the rest of the chapter, the regularization is taken to be ℓ_2 norm by setting $\lambda_1 = 0$ (Ridge). The analysis with ℓ_1 norm regularization (LASSO) by setting $\lambda_2 = 0$ to enforce sparsity is discussed in *SI Stability of Analysis*.

The CNNs used are pre-trained CNNs for classification tasks. They are trained based on a 1000-object classification task on the ImageNet dataset from the ImageNet Large Scale Visual Recognition Challenge [172]. One legitimate concern of deploying neural networks in modeling is that interpretations about the models may depend on the details of the neural network architecture choices. To address this problem, we use three different neural network architectures to model V4 neurons: AlexNet [101], GoogleNet [183] and VGG [179]. All three networks have high classification performance on ImageNet recognition challenge and are known to provide transferable image features in other computer vision tasks such detection and segmentation [177, 205]. To vary the number of layers, we use features from layer two, three and four of each network. Later in this section, we show that using layer 1 and layers higher than layer 4 leads to lower

prediction accuracies or has too large receptive fields not comparable with those of V4 neurons. Finally, on top of the CNN features, either Ridge or Lasso regression is used to predict the (spike) firing rates. As a result, we obtain 18 models for each neuron (3 nets $\times$ 3 layers $\times$ 2 regression models). Next we provide detailed prediction performance of these 18 models and compare them to previous models in the literature before we propose the stability-driven interpretation and visualization framework of DeepTune based on a stable aggregation of all 18 models.

To determine quantitatively how well our models describe the responses of each neuron, we test their performance on the holdout test set. All our models were estimated using the training data set. The correlation between the firing rates predicted by the model and the actual average firing rates on the test set is used as the prediction performance for all our 18 models. As a baseline for comparison, we also fit a V1-like Gabor wavelet model [45, 93]. The Gabor wavelet model first extracts image features by applying a bank of linear Gabor wavelet filters to the input image at varying orientations, spatial frequencies and phases, followed by half-wave rectification and a compressive nonlinearity, then regresses the responses of each neuron using Ridge regression [78].

Our AlexNet-Layer2 (+Ridge) model has a average correlation coefficient of 0.44 (or 0.52 for noise-corrected correlation coefficient [176]) on the holdout test set. It achieves the state-of-the-art prediction accuracy for V4 neurons on natural image stimuli [46, 201]. Comparing to [46], our average correlation coefficient is about 0.15 higher. As shown in Figure 6.2-D, all of the 18 models have average correlation coefficients higher than 0.42. For nearly all of the 71 V4 neurons, they are all more accurate than the V1-like Gabor wavelet model (with an average correlation coefficient 0.33). Due to space limitations, we plot the results only for 4 models, which are all based on AlexNet-Layer2, AlexNet-Layer3, VGG-Layer2, GoogleNet-Layer2 (and ridge) in Figure 6.2-A and 6.2-B. The first two models are chosen in order to demonstrate stability of prediction results and interpretations across different CNN layers, while the other two models are chosen to show stability across different CNN architectures. In Figure 6.2-C, we compare the average prediction performance for models from all 7 layers of AlexNet for 71 neurons. The model based on AlexNet-Layer1 has similar performance to that of the V1-like Gabor wavelet model; while models from layers 2 to 5 have much higher predictive performance (e.g. 0.44 for layer 2, 0.46 for layer 5). This justifies the recent finding [177] that the intermediate layers of pre-trained CNNs (on large-scale image classification tasks), like AlexNet, can extract more complex features than the first layer and Gabor wavelets.

In order to be consistent with the literature [168, 201, 46], we also report the proportion of explainable variance captured by a model. It attempts to control for differences in noise levels between experimental setups, individual neurons, and brain regions. We estimate the explainable variance through the noise-corrected correlation coefficient [176] using the repeated data in the holdout set (see *SI Methods* for more information). Averaged over the 71 V4 neurons, the AlexNet-Layer2 and ridge model captures 30.3% of the explainable variance. This performance matches the 30% of computational models

for area V2 [196]. The unexplained portion of the response is very likely to have resulted from two factors: visual tuning properties not described by the AlexNet-Layer2 (and ridge) model and non-stimulus influences on the response. The latter is unlikely to be removed completely given our experimental setups [196]. Note that the prediction task on the natural images in this chapter is substantially harder than that on images with artificial objects overlaid in [204]. Besides this work [204] on simpler natural image stimuli, our CNN-based models demonstrate a large improvement in prediction performance over previous works with natural image stimuli similar to ours [46, 201]. In the next section, we take advantage of this high prediction accuracy to better characterize of V4 tuning properties via DeepTune images.

6.4 DeepTune as a naturalistic visual representation of tuning

It has long been challenging to fully characterize shape tuning properties in area V4. There are two main difficulties: the absence of highly predictive and biologically plausible computational models for the nonlinear response properties of V4 [169], and the lack of systematic methods to generate relevant complex natural stimuli to probe V4 neurons more efficiently. Given the state-of-the-art predictive performance of our CNN-based models, it is natural to ask whether these models could also provide a better characterization of shape tuning (e.g. angular, curvature or orientation tuning) or texture tuning in area V4. However, unlike existing studies using relatively simple Gabor wavelets [45, 93] or Fourier transform [46], complex nonlinear CNN features in our models make it extremely challenging to consistently interpret our models.

Inspired by computer vision advances in visualizing CNNs [208, 125], we introduce *DeepTune images* as a naturalistic visual representation of tuning for a V4 neuron. The DeepTune images are made of a collection of reconstructed images that jointly represent the shape tuning properties of a neuron. For each neuron and for each given model, a *DeepTune image* (or preferred DeepTune image) is obtained by optimizing over the input image space to maximize a regularized model output (predicted neuron response). Starting from a random image (e.g. white noise image with zero mean and fixed small variance), we use the gradient ascent method to gradually increase the model output until convergence. Formally, given a fixed predictive model at a particular time lag (the single lag time that causes best prediction performance in a 10% validation set split of the training set) $f : \mathbb{R}^{s\times s} \mapsto \mathbb{R}$, we seek an input image $\mathbf{z} \in \mathbb{R}^{s\times s}$ that minimizes the following objective function:

$$-f(\mathbf{z}) + \lambda_p \mathcal{R}_p(\mathbf{z}) + \lambda_{\text{TV}} \mathcal{R}_{\text{TV}}(\mathbf{z}).$$

The regularization terms are included to capture prior information about natural images. That is, the optimization search is constrained to be close to the set of smooth and naturalistic images [125]. The specific regularization choices above are motivated

by image denoising techniques [171] and by natural image statistics [178]. The first regularizer $\mathcal{R}_p$ (the ℓ_p-norm of a vectorized image pixels) encourages the intensity of pixels to stay small. By choosing a large p ($p = 6$ in our analysis), this regularizer prevents the solution image from taking extremely large pixel values. The second regularizer $\mathcal{R}_{\text{TV}}$ controls the total variation norm of an image. It encourages the image to be smooth and removes excessive high-frequency details (see *SI Methods* for more information).

The collection of DeepTune images is constructed from all 18 predictive models. In addition, we verify that 10 independent random initializations of starting images do not change the output much (see *SI Stability of Analysis, Figure S7*). Similarly, an inhibitory DeepTune is obtained by minimizing instead of maximizing the model output. We note that the DeepTune images differ from the traditional receptive fields in neurophysiology [85, 45] in two ways: multiple images are used to describe tuning properties of a single neuron; they are more naturalistic representations of tuning with a higher resolution.

Figure 6.3-A shows the DeepTune images from 4 of our 18 models built for Neuron 1. We visually observe that these DeepTune images share a stable curvature pattern with edges forming an angle of nearly 90 degrees. The rest 14 DeepTune images produced from the other 14 models differ slightly, but the main curvature pattern remains relatively stable (see *SI Stability of Analysis Figure S8*). That is, the curvature angle stays close to 90 degrees and the spatial location of the curvature pattern remains at left side of the image. To further quantify the curvature angle and spatial frequency, we compare the power spectral densities (PSD) of these DeepTune images in Figure 6.3-B. All four DeepTune images share a strong and stable frequency component in the range of 45 to 135 degrees with spatial frequencies of 2 to 5 cycles per receptive field (green). Note that the high frequency components from the Model-4 DeepTune image are not consistent with the other three models. Especially, GoogleNet-Layer2 model has high frequency components that are not present in three other models. Therefore these components likely reflect noise and should be discounted. In Figure 6.3-C, we visualize the spectral receptive field (SRF) model [46] for Neuron 1. The SRF visualization shows the frequency components of the stimulus image selected by SRF model. The color map (red-blue) is chosen to be different from that of the DeepTune Fourier transform (green-pink). The color map difference serves a reminder of the difference between PSD and SRF. As observed from the DeepTune image PSD, the SRF model also shows that Neuron 1 exhibits a strong preference to the frequency component in the range of 45 to 135 degrees with spatial frequency of 2 to 5 cycles per receptive field. In addition to DeepTune and SRF, this curvature tuning is further supported by the curvature patterns in the images from training and test sets with the highest responses for Neuron 1 (Figure 6.3-D and E). Figure 6.3-E illustrates the measured and predicted firing rates in test set from the 4 models as well as the predicted firing rates from the SRF model. For this Neuron 1, our 4 models have similar prediction accuracies (correlations on the holdout set between 0.61 to 0.64), while the SRF model has difficulty capturing the peak firing rates as seen in the lower plot of Figure 6.3-E, with a corresponding

correlation of 0.42.

In addition to the visual comparison of 18 distinct DeepTune images generated from 18 models, we introduce consensus DeepTune to capture in a single image the stable patterns across 18 models. The consensus DeepTune image is obtained via a similar optimization scheme as in the original DeepTune optimization for a single model, but with an aggregation of gradient information from all 18 models. The aggregated gradient maintains the stable components in the gradients and discounts the unstable components (more details in *SI Methods*). Both excitatory and inhibitory consensus DeepTune images for Neuron 1 are shown in Figure 6.3-F. The excitatory consensus DeepTune (Figure 6.3-F) exhibits curvature contour patterns that visually matches all 4 models (Figure 6.3-A). The power spectral density (PSD) to the right of the consensus DeepTune image in Figure 6.3-F similarly matches the individual models. This PSD displays strong frequency components in the range of 45 to 135 degrees with spatial frequencies of 2 to 5 cycles per receptive field. On the other hand, the inhibitory consensus DeepTune consists of lines orthogonal to the curvature contour (see *SI Stability of Analysis* for comparison with inhibitory DeepTune images from all 18 models). Some blobs are also visible in the inhibitory consensus DeepTune image, suggesting that the response of Neuron 1 is attenuated by blob-like texture patterns. This is further supported by observing that the inhibitory PSD contains strong high frequency components on the top center.

The consensus DeepTune image captures the stable components of DeepTune images across our 18 models. It can be visually observed that the DeepTune images from a number of individual models are very similar to the Consensus DeepTune (see *SI Stability of Analysis, Figure S8*). To quantify this similarity, we compute the Pearson correlation coefficient between pixel values of the consensus DeepTune and those of each DeepTune image. Figure 6.3-G visualizes boxplots of these correlation coefficients. Each boxplot corresponds to one of the 18 models and shows the distribution of 71 correlation coefficients for all 71 neurons for this model. The median correlations for all of the models are considerably high. The highest median correlation is 0.83 which is achieved by AlexNet-Layer2 and GoogleNet-Layer3 with ridge regression. Models with lasso tend to have lower similarities to the consensus DeepTune. Due to space limitations, in the subsequent sections we present by default the consensus DeepTune image as a stable representation of a V4 neuron's tuning property. Although a single consensus DeepTune image seems to be sufficient, the stability analysis across 18 DeepTune images are necessary to determine the spatial locations of the stable parts. This is to ensure that we identify only the stable locations of the consensus DeepTune image to be interpreted.

6.5 Model-selected CNN features highlight receptive fields

DeepTune images can be generated for any black-box predictive model for which the gradient computation is not difficult. However, our model is not just a black-box: when we used the convolutional filters for feature extraction, we implicitly assumed that the V4 feature extraction happens using successive linear and non-linear combination of spatially localized low level filters. In this section, for each neuron, we visualize the low level filters that played an important role in building the model. This filter visualization would allow us to relate to previous literature on the spatial receptive fields of V4 neurons.

Taking AlexNet-Layer2 model as an example, we examine its regression weights (see *SI Stability of Analysis, Figure S11* for visualization of weights from other models). Regression weights with large magnitudes indicate high sensitivity of the neuron to particular image features. The AlexNet-Layer2 features are of dimension $256 \times 13 \times 13$. They consist of 256 different convolutional filters that are spatially located on a grid of size 13×13. The corresponding regression weights at one time lag is of the same dimension. We examine the regression weights by asking the following two questions: where on the image are the regression weights with the largest magnitudes? What kinds of convolutional filters contribute the most to the prediction performance?

To answer the first question, we define an *average regression weight map* as the sum-of-squares pooling of regression weights on the CNN features. It is defined across the different convolutional filters and the time lags at each location on the 13×13 spatial grid. Formally, for each neuron, let $\hat{\boldsymbol{\beta}}_{mijk}$ be the regression weight for filter m at spatial location (i, j) and lag k. Then the average regression weight map $\Phi \in \mathbb{R}^{13\times13}$ is defined as follows:

$$\Phi_{ij} = \sum_{m=1}^{256} \sum_{k=1}^{k} \hat{\boldsymbol{\beta}}^2_{mijk}.$$

Figure 6.4-A shows the average regression weight map from the AlexNet-Layer2 model for 4 neurons. On the 13×13 grid map, lighter pixel color indicates higher weight map value. Maps from other models share stable shape and location (see *SI Stability of Analysis, Figure S11* for a comparison across models). For each neuron, the average regression weight map presents an estimate for the spatial receptive field. Maps for V4 neurons exhibit diverse shapes. For example, the receptive fields for Neurons 1 and 2 have round shapes, while those for neurons 3 and 4 form straight or curved band shapes. These CNN-based spatial receptive fields provide an alternative to [140] for showing diversity in the size and shape of the receptive fields of V4 neurons. These regression weight maps are also indicative of the regions where DeepTune images across 18 models share stable patterns. Figure 6.4-B displays the DeepTune images from the AlexNet-Layer2 model for the 4 neurons, along with the consensus DeepTune images in Figure 6.4-C. The corresponding inhibitory DeepTune image and consensus

inhibitory DeepTune image are shown in Figure 6.4-D and E respectively. Looking at the patterns of the DeepTune images, Neuron 1 is tuned to the curvature-contour shapes with edges forming an approximately ninety-degree angle. Neuron 2 is tuned to blob-like patterns and textures. Neuron 3 is selective to curvature patterns with a strong diagonal line preference and Neuron 4 is tuned to corner-like shapes with edges forming ninety-degree angles. The tuning patterns shown via DeepTune are consistent with receptive field shapes shown in regression weight maps.

The second question is: which types of convolutional filters contribute the most to the prediction performance? To address this question, we quantify the importance of each convolutional filter by ℓ_2 pooling of the regression weights for a convolutional filter across spatial locations. Formally, for each neuron, the filter importance I_m of m-th convolutional filter is defined as follows,

$$I_m = \sum_{i=1}^{13}\sum_{j=1}^{13}\sum_{k=1}^{k} \hat{\boldsymbol{\beta}}^2_{mijk},$$

where $\hat{\boldsymbol{\beta}}_{mijk}$ is defined as before. This filter importance index provides an independent view of neuron shape tuning through the most and the least important filters. To interpret the filter importance, a visualization of each convolutional filter in CNN is required. To this end, we adopt the filter visualization technique introduced by [208]. For each filter, we show the 9 top image patches from the ImageNet training set that have the highest filter responses (see *SI Methods* for visualization of AlexNet filters). These 9 top image patches are representative of what this convolutional filter is computing [208, 206]. Taking Neuron 1 as an example, Figure 6.4-F and G show the top and bottom two filters among 256 filters in AlexNet-Layer2 model ranked by the filter importance index, I_m.

For each neuron, we observe that the top two filters capture essential image components corresponding to the tuning patterns shown in the DeepTune images. These tuning patterns are long curvatures for Neuron 1, blob-like patterns for Neuron 2, diagonal lines for Neuron 3, and corner-like shapes for Neuron 4. Comparing to the DeepTune images (Figure 6.4-B-C-D-E), the most important and least important CNN-features (Figure 6.4-C-H-I) provide an alternative interpretation of the excitatory and inhibitory tuning property of V4 neurons, respectively. Figure 6.4 shows that these two views (I_m based and DeepTune) are visually consistent.

6.6 The wide variety of shape and texture tuning in V4

From the four DeepTune visualization examples above, we observe that both V4 neurons selective to curvature or to texture are present. Previously, on the one hand, V4 neurons are shown to be tuned for orientation and spatial frequency of edges and linear sinusoidal gratings [49], non-Cartesian gratings [64, 65] and curvature of contours [154, 153]; on the

other hand, V4 is believed to play a major role in processing textural information [133, 7, 151]. We investigate the distribution of these two categories of neurons through cluster analysis and DeepTune visualization.

Since our interpretation of the model highly depends on how accurate the model describes the stimulus-response relationship, we filtered out 25 neurons with correlation > 0.5 from all 71 neurons for the cluster analysis (see *SI* for the cluster analysis on all 71 neurons). Based on the feature importance (I_m) defined in the previous section, we clustered these 25 neurons via hierarchical clustering with euclidean metric and single linkage. From the dendrogram (Figure 5), four clusters are obtained by cutting off around the root level. In principle, it is hard to visualize the meaning of the clusters in such a cluster analysis, the DeepTune images allow us to understand these cluster in more details. Figure 5-B shows that the four clusters consist of One cluster for local corner/blob texture, long curved contours, V1 like long edge pattern and very complex patterns. About 40% of them are selective to textures, 30% of them prefer shapes and the other 30% are either selective to complex patterns or simple V1 like patterns (see *SI* for the result on the all 71 neurons, where the proportion is similar). The cluster analysis with DeepTune visualization extend the results in previous studies on V4 neuron selectivities[65, 99] by displaying neuron tuning in a concrete and naturalistic manner.

6.7 V4 curvature tuning to a full range of separation angles

It is suggested by Roe et al. [169] that diverse curvature tuning in V4 provides an efficient way to encode shapes. However, it is not yet clear that how different types of curvature tunings are distributed in the V4 population. Previously, artificial curvature stimuli have been used to probe the different angle tuning properties in area V4 [154, 153]. These stimuli are constructed by joining two oriented line segments in a sharp corner or curve. These studies highlight the presence of bimodal orientation tuning with various separation angles. The preferred separation angle is defined in [153, 46] as the angle between the two most preferred oriented line segments passing through the center. The SRF analysis [46] also confirm bimodal orientation tuning in V4 by showing the presence of neurons tuned sharp corners. As for the distribution of different angles, Carson et al. [30] observed that not all curvatures are equally represented. They use sparse modeling of object coding to show that the strong representation of acute curvatures across the neural population. In this section, we investigate whether DeepTune images can concretize previous discoveries and provide visualization of V4 neurons tuned to different separation angles.

By visually inspecting the consensus DeepTune images of 71 V4 neurons, we first identified the 38 neurons that are tuned to curved contours, corner-like shapes and lines. Then we manually clustered these 38 neurons into four categories based on their

separation angles of their curves (45°, 90°, 135°, 180°). Figure 6.6-A shows a count histogram of (excitatory) separation angle of the 71 V4 neurons. We observe that there is a strong presence of neurons with curvature tuning at less or equal to 90° separation angles (18 out of 71 neurons). Another 15 neurons are selective to blob-like textures that does not correspond to any particular angle. There are 18 neurons that are not selective to any clear angle or blob-like patterns.

To further support the separation angles for V4 neurons identified by looking at DeepTune images, we perform spectral receptive field (SRF) analysis [46] on our data and compare the angles identified by both analyses. In Figure 6.6-B and C, for each neuron, we display in one column the consensus DeepTune image and the SRF plot as in David et al. [46]. The horizontal axes of the SRF show the orientation tuning of each neuron, with preferred component in red. In the SRF plot, according to [46], the separation angle corresponds to the difference between the top two orientation tuning peaks. We observe that the separation angle from the SRF plot are consistent with the ones from the DeepTune images. For example, for the bottom left neuron, both DeepTune and SRF show two orientation tuning peaks at about 70° and 120°. To summarize, the diversity of excitatory curved-contour patterns in fact matches the previous neurophysiological observations in V4 [153, 30, 140]. Furthermore, our DeepTune images offer a concrete visualization of the bimodal orientation tuning properties of many V4 neurons, refining earlier analysis.

6.8 Suppressive tuning discovery via inhibitory DeepTune

It is well known that V4 neurons have surround suppressive mechanisms [49, 174, 100] just like many other visual cortical areas [85, 5]. Besides, recent study by Willmore et al. [196] found evidences for the presence of strong suppressive tuning to specific features in about half of the neurons in area V2. In addition, they show that this type of suppressive tuning is not caused merely by surround suppression and is not present in area V1. In this section, we investigate whether such strong suppressive tuning is also present in area V4.

To study the suppressive tuning in the area V4, we fit the Berkeley Wavelet Transform (BWT) model [196] to our data. The BWT-based model provides a nonlinear spatio-temporal receptive field (STRF) for each neuron. We adopt the excitation index (EI) introduced in [196] as:

$$EI = \frac{\Sigma h^+ - \Sigma h^-}{\Sigma h^+ + \Sigma h^-}$$

where h^+ and h^- are positive and negative weights respectively assigned to the wavelets in each STRF.

The BWT-based model has an average prediction correlation coefficient 0.33 for the 71 V4 neurons in the holdout test set. It is about 0.09 lower than the worst among 18

CNN-based models. While this model does not fully explain the non-linear property of V4 neurons, its accuracy is comparable to that of the same BWT model for V2 neurons (average correlation coefficient of 0.30) [196]. Figure 6.7-A shows the histogram of excitation index for 71 V4 neurons. 41% of the neurons in V4 show suppressive tuning. The median of the excitation index for V4 neurons is 0.10. While the portion of neurons with suppressive tuning is 9% lower compared to that in V2, it is 29% higher than that in cortical area V1 [196].

Figure 6.7-B presents the excitatory and inhibitory consensus DeepTune images for three neurons identified as suppressive neurons according to the BWT model (on the left side of the histogram). The corresponding excitation indexes are shown below the neuron names. Recall that the excitatory DeepTune images are obtained via maximizing the model response (with appropriate regularization), while the inhibitory ones are obtained via minimizing the model response (with appropriate regularization). The neuron excitation index and response of the model to each DeepTune image are shown in the same panel. For example, $\hat{y} = 0.54$ means that the model predicted a firing rate of 0.54 spikes per sampling period (16.7ms). The DeepTune images provide a concrete visualization of the suppressive tuning in V4: The excitatory DeepTune images of these neurons are weak and/or blurry, while the inhibitory DeepTune images show sharper patterns. In the case of Neuron 43, while the excitatory DeepTune has blurry patterns, the inhibitory DeepTune exhibits a clear tuning to ninety-degree corner shapes in the right hand side of visual field. This means that a ninety-degree corner shape is likely to drive this neuron firing rates close to zero. Moreover, looking at the other inhibitory DeepTune images, both of neurons 27 and 26 have strong suppressive tuning to complex shapes with mid-range frequency.

6.9 Summary and discussion

In this chapter, we demonstrated that models combining pre-trained CNN features with regularized linear regression lead to state-of-the-art results in modeling V4 neuron responses to natural images. More importantly, we introduced DeepTune images that reveal fundamental properties of V4 encoding. In particular, we find that individual V4 neurons exhibit both tuning for shape and texture and many V4 neurons are highly selective for complex patterns that are difficult to describe in words.

6.9.1 Flexible visualization of optimal stimuli

The idea of computationally optimizing input stimulus to discover neuron tuning properties dates back to Carlson et al. [30]. The evolutionary sampling method was used to optimize for the stimulus that causes the highest number of spikes. This work greatly expanded the search space of tuning patterns compared to previous methods that were based on handcrafted stimuli [154, 64]. However, the evolutionary sampling method in [30] is constrained on limited concatenated Bezier splines. It can generate spline-

based contours easily, but has difficulty for generating fine-scale texture stimuli. Our DeepTune images are generated from a regularized optimization directly over the input pixel values, and hence have an even larger search space that allows for more complex and naturalistic tuning patterns.

The resulting DeepTune population analysis demonstrates that V4 neurons are tuned to a huge variety of shapes as well as textures in different orientations. It also reveals that the tuning properties of many V4 neurons cannot be explained by simple edge and corner patterns. We see in Figure 7, for example, that even the stable part of the DeepTune images is difficult to describe in such simple terms. This suggests that tuning in area V4 is much more complex than that of V1 and than what can be described by handcrafted grating stimuli. Studies based on synthetic stimuli [64, 65, 154] may lack the expressive power to represent shapes of many V4 neuron receptive fields. Predictive modeling approaches as SRF [46] may not be sufficient to capture the complex tuning properties either. It provides only summary statistics such as spatial frequency and orientation about the receptive fields.

6.9.2 Distinctions in curvature selection revealed by DeepTune

Examining the DeepTune images of Neuron 3 and 4 in Figure 4, we see that both neurons are tuned to curvatures with similar edge orientations (two edge directions with a separation angle of ninety degrees). However, they have very distinct shape tuning properties apart from the orientation tuning summary statistics. Neuron 3 prefers a curvature-contour pattern with a ninety-degree angle and long edges. Neuron 4 prefers a corner-like repeated texture. This agrees with the study by Nandy et al. [140]. It is suggested that the curvature selection of V4 neurons could arise for two reasons: systematic variation in fine-scale orientation tuning across spatial locations (like Neuron 3), and local tuning heterogeneity (like Neuron 4). Note that this type of refined result would be difficult to obtain via methods based on global Fourier analysis such as spectral receptive field (SRF) [201, 46]. The 2D Fourier transform is spatial translation-invariant, meaning it is difficult to distinguish between Neuron 3 and Neuron 4 via SRF analysis.

6.9.3 DeepTune for future neurophysiology experiments

The DeepTune images for each V4 neuron are concrete and naturalistic. They are visually very similar to many input image stimuli. In other words, the DeepTune images are ready to be fed back to neurons as stimuli for confirmation or refutation of their characterizations of tuning properties in a closed experimental loop. Consequently, DeepTune images hold the promise to speed up the efficiency of data collection in V4 and other brain areas.

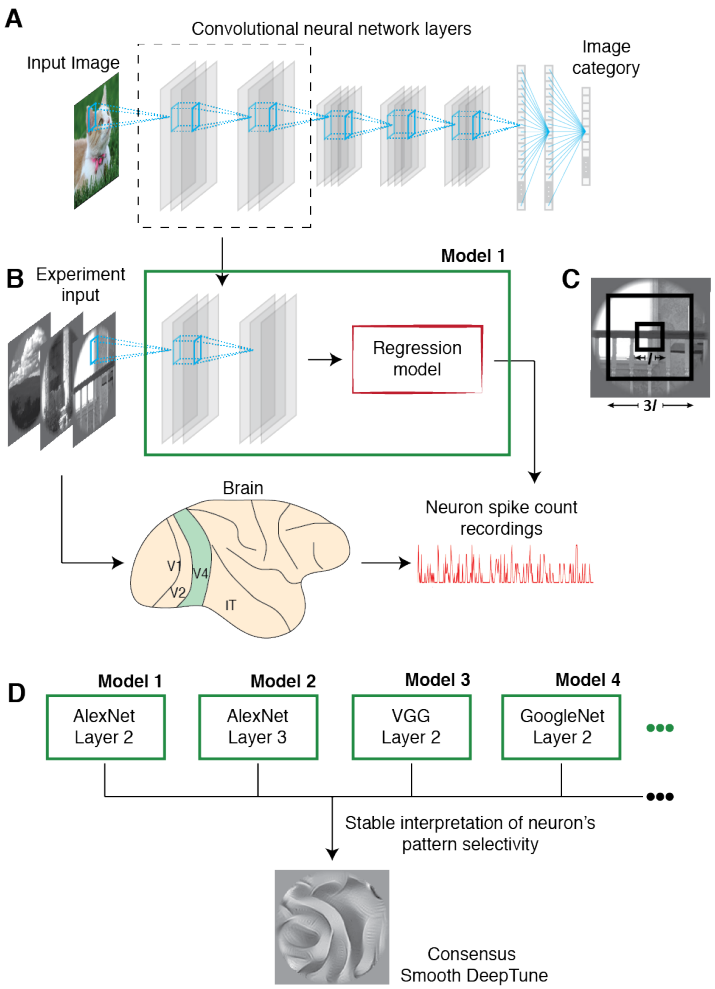


Figure 6.1. DeepTune framework through transfer learning: first, we use features from pre-trained convolutional neural networks (CNNs) in regularized regression to predict (spike) firing rates of neurons in the visual area V4; second, stability-driven DeepTune images across 18 CNN-based predictive models are generated for interpretation. **A.** Architecture of a convolutional neural network (CNN) pre-trained to perform 1000-class image classification task on the ImageNet dataset (e.g. AlexNet). **B.** An input image is propagated forward in a fixed layer of the CNN, yielding a feature vector representation of the image. This vector is used to fit a regularized linear regression model to predict firing rates of each V4 neuron. **C.** The classical receptive field (CRF) during the experiment is set in the middle of the stimuli with width l while the whole image has the width $3l$. **D.** 18 accurate predictive models are obtained using features from layers 2, 3, 4 of three pre-trained AlexNet, GoogleNet, VGG, with either ℓ_1 (lasso) or ℓ_2 (ridge) regularized linear regression. DeepTune, a stability-driven interpretation and visualization framework of CNN-based model (across multiple such models) is proposed to characterize V4 neurons' tuning preferences (more details in ??????). The consensus DeepTune image for one neuron (corresponds to Neuron 1 in Figure 6.3-A) is shown and displays a stable curvature pattern with edges forming an approximately ninety-degree angle.

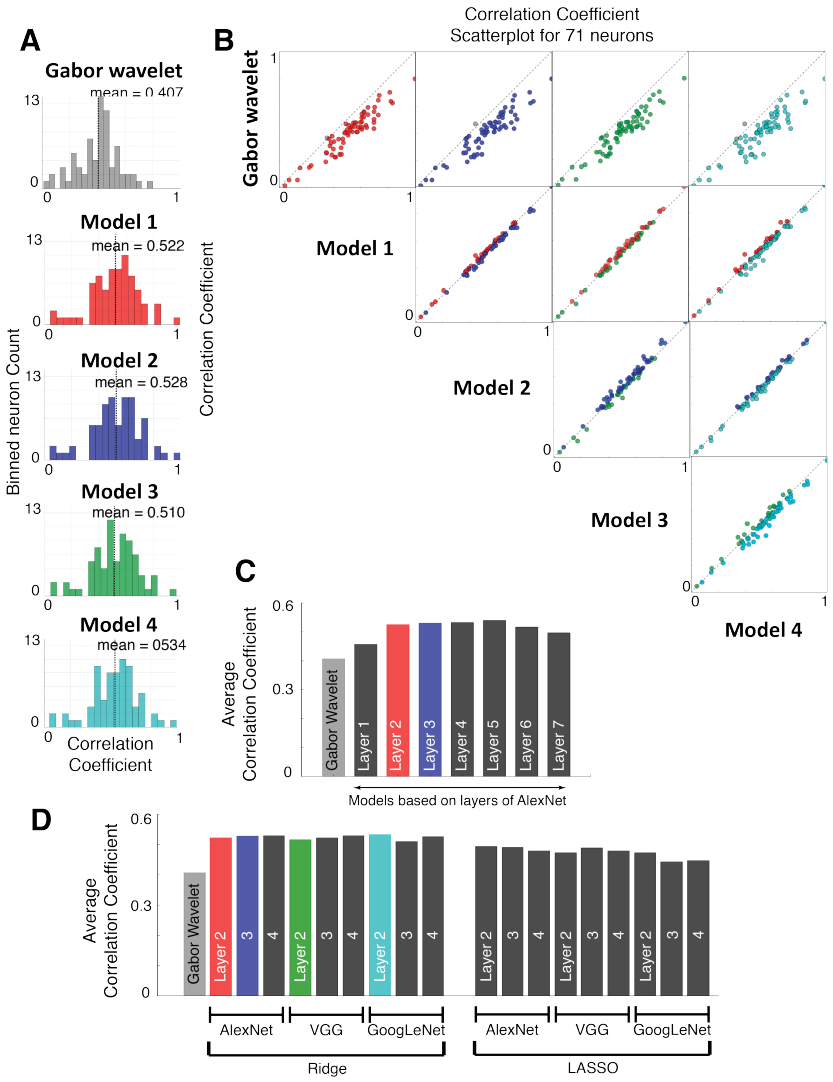


Figure 6.2. CNN-based models outperform a V1-like Gabor wavelet model in terms of noise-corrected correlation coefficient [176] as the prediction performance measure. **A.** Histogram of noise-corrected correlation coefficients over the population of 71 V4 neurons for 4 models are shown, where the baseline model is a V1-like Gabor wavelet model, Model 1 corresponds to AlexNet-Layer2, Model 2 AlexNet-Layer3, Model 3 VGG-Layer2, and Model 4 GoogleNet-Layer2. Ridge regression is used in all 4 models. **B.** Scatter plots comparing noise-corrected correlation coefficients of 71 neurons between each pair among Models 1-4. **C.** Average prediction performance across 71 neurons for models from all 7 layers of AlexNet with ridge regression. The model based on AlexNet-Layer1 has the closest performance to that of the V1-like Gabor wavelet model; while models from layers 2 to 5 have higher predictive performance. **D.** Average prediction performance across 71 neurons for all 18 models. All 18 models perform similarly in prediction and much better than the Gabor wavelet model and the ridge-based models perform overall better than the lasso-based ones. Moreover, higher layers and more complex CNNs seem to result in worse performance for lasso, but not for ridge.

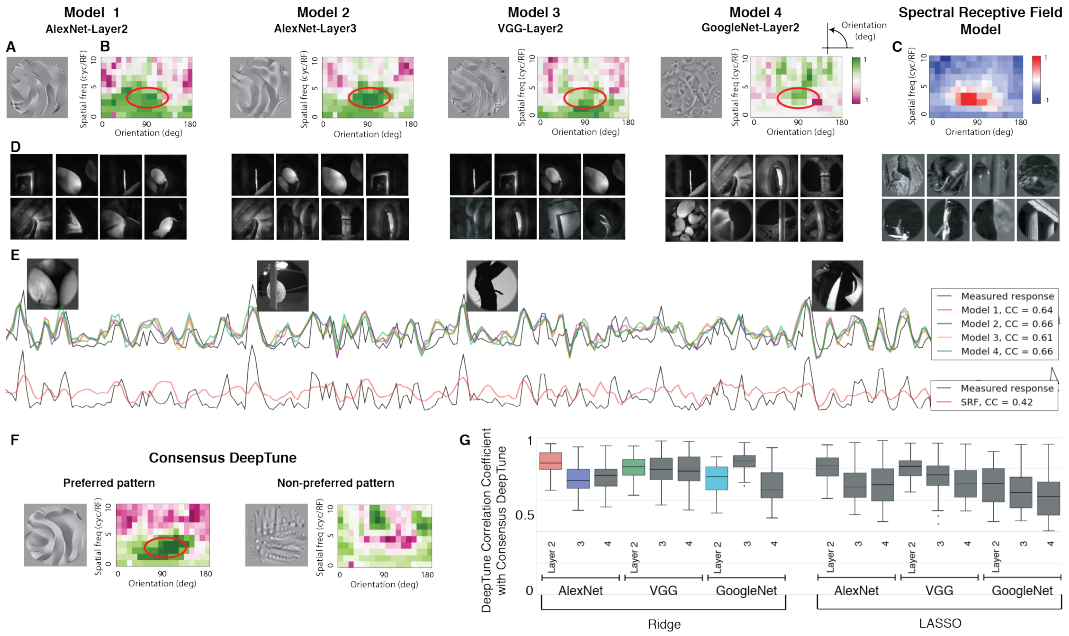


Figure 6.3. DeepTune images from four of our 18 models built for Neuron 1. **A.** DeepTune images based on Models 1-4 for Neuron 1. These images share a visually stable curvature pattern with edges forming an approximately ninety-degree angle. **B.** Power spectral densities (PSDs) of the DeepTune images in polar coordinates. Through the PSDs, all four DeepTune images share a strong and stable frequency component in the range of 45 to 135 degrees with spatial frequency of 2 to 5 cycles per receptive field (the green color). **C.** Visualization of spectral receptive field (SRF) [46] model for Neuron 1. The SRF visualization emphasizes in red the frequency components of the stimulus image selected by the SRF model. The pattern selectivity according to SRF is consistent with the stable part of the PSDs of DeepTune images (highlighted in red circles). **D.** Images from training set with the highest responses for Neuron 1. Similar curvature patterns to the DeepTune visualization are visible in these images. **E.** The measured and predicted (spike) firing rates in the test set from Models 1-4 as well as the SRF model for Neuron 1. Images from the test set with the highest responses are visualized on top of the corresponding spike rate. Similar curvature patterns are visible in these images. Correlation coefficients between the measured and predicted firing rates are shown in the right panel. All four models outperform the SRF model. **F.** The consensus DeepTune image for Neuron 1. Both excitatory, inhibitory DeepTune images and the corresponding PSDs are shown. The excitatory pattern based on the consensus DeepTune exhibits the curvature contour that is similar to those from the four models in panel A. The inhibitory pattern visually consists of lines orthogonal to the preferred curvature contour, confirmed via PSD visualization on the right. **G.** Each box-plot corresponds to a CNN-based model among the 18 models and is based on 71 raw-pixel correlation coefficients. Each such coefficient corresponds to a neuron and is calculated between the consensus DeepTune image and a DeepTune image from that model and for that neuron. DeepTune images from AlexNet-Layer2 and GoogleNet-Layer 3 have the highest similarity on average to the consensus DeepTune image.

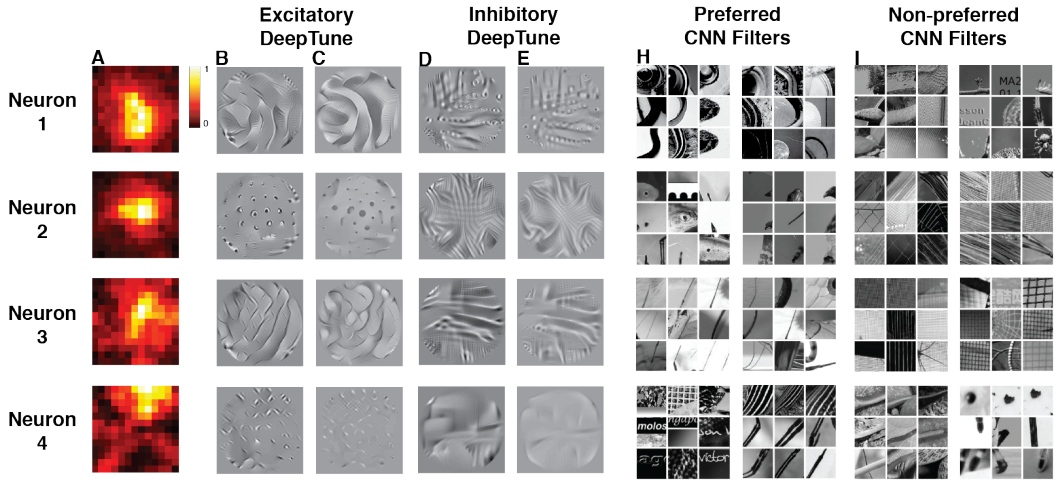


Figure 6.4. For Neurons 1-4, a comparison of excitatory and inhibitory DeepTune images, average regression weight maps and selected CNN features. **A.** Average regression weight map based on the AlexNet-Layer2 model. For each neuron, the average regression weight map also exhibits stable patterns across models (see *Stability of Analysis, Figure S11*) and it highlights the receptive field of a neuron. **B.** Excitatory DeepTune images from the AlexNet-Layer2 Model. Neuron 1 is tuned to the curvature-contour shapes with edges forming an approximately ninety-degree angle. Neuron 2 is selective for blob-like patterns and textures. A DeepTune image for Neuron 3 shows selectivity to curvature patterns with a strong diagonal line preference. Neuron 4 is tuned to corner-like shapes with edges forming ninety-degree angles. The rest of the 17 models show consistent patterns as shown in other DeepTune images (see *SI Stability of Analysis, Figure S8*). **C.** Excitatory consensus DeepTune images based on all 18 models. **D.** Inhibitatory DeepTune images from the AlexNet-Layer2 Model. **E.** Inhibitory consensus DeepTune images based on all 18 models. **H.** Top two excitatory CNN filters based on the filter importance index. To visualize a convolutional filter from a CNN, the 9 top image patches are presented from the ImageNet training set that have the highest filter responses. These 9 top image patches are representative of what this convolutional filter is computing [208, 206]. The top two selected CNN filters support the findings based on DeepTune images. For example, Neuron 1 is tuned for curved-contour patterns according to DeepTune images and its top CNN filters are those that activate on curvatures of similar shapes. Neuron 2 is selective for blob patterns and the top CNN filters activate respectively on blob pattern or pieces of a blob pattern. **I.** Top two inhibitory CNN filters based on the filter importance index.

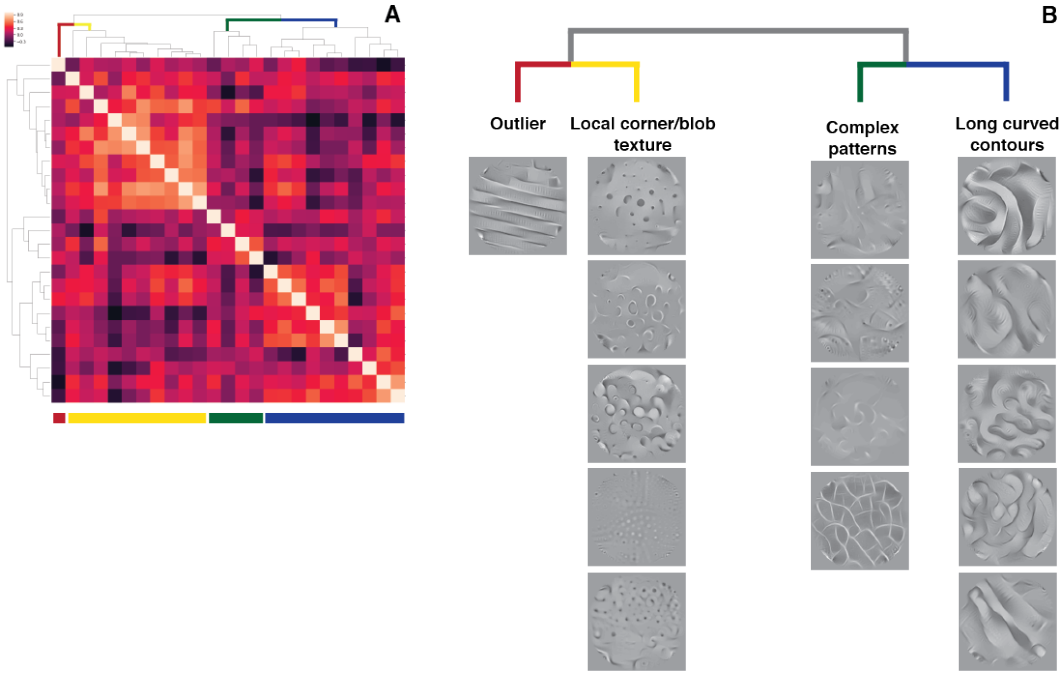


Figure 6.5. Diversity and clustering among 71 V4 neurons. Neurons are manually categorized into four categories by applying hierachical clustering on feature importance I_m. **A.** Correlation heatmap based on feature importance I_m and hierachical clustering with euclidean metric and single linkage. **B.** Four clusters of V4 neurons. More than 40% of the neurons are selective to texture, half of which prefer blob-like textures and the other half prefer corner-like textures. About 30% of the neurons exhibit contour patterns, both curvature and straight lines. The rest of the neurons have selectivities to visually complex patterns.

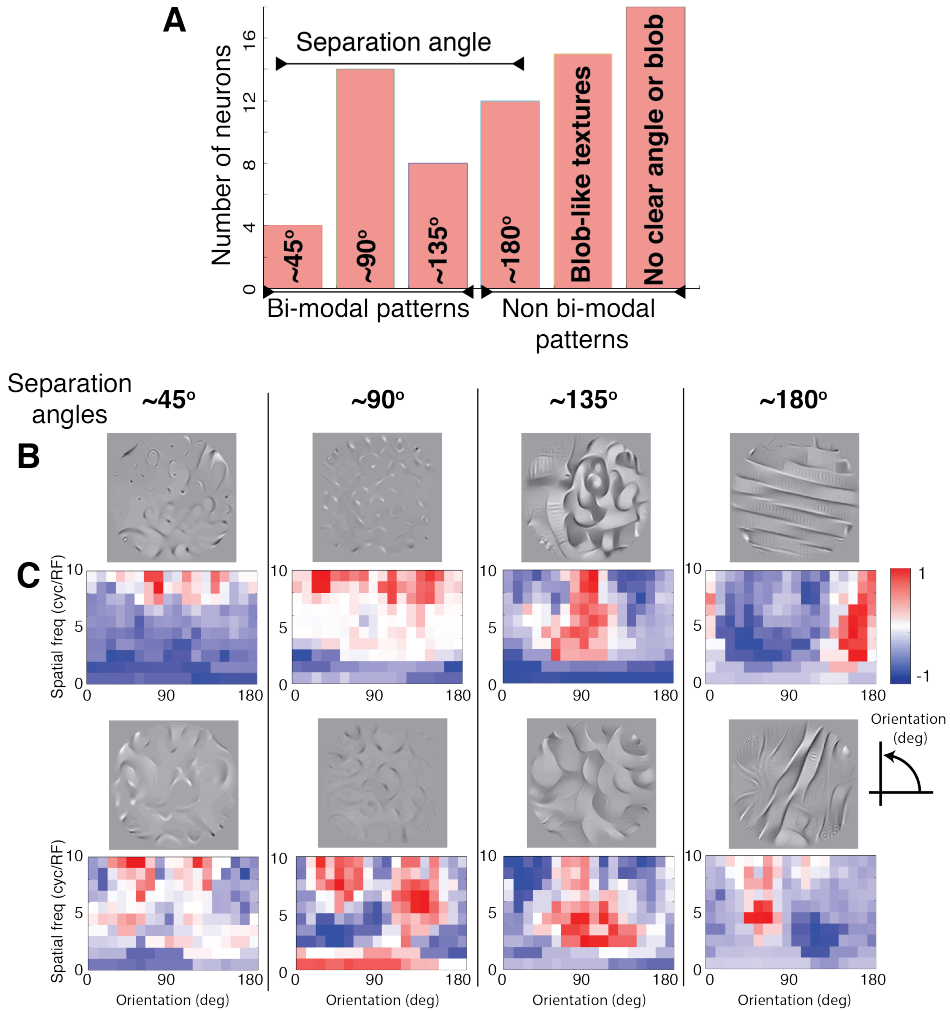


Figure 6.6. Categorization of V4 neurons based on their separation angles. **A.** Neurons are manually categorized into six groups. The first four groups contain neurons tuned to patterns with separation angles of 45°, 90°, 135°, and 180°. These patterns are either contours or textures. About 20% out of 71 neurons are tuned to patterns with separation angles close to 90°. Another 20% of the neurons are selective to blob-like textures that do not correspond to any particular angle. The rest of neurons are not selective to any clear angle or blob-like patterns. **B.** The consensus DeepTune images for two example neurons in each of the first four categories. **C.** The corresponding spectral receptive field (SRF) (David et al [46]) visualization. The orientation tuning obtained via SRFs and consensus DeepTune images are consistent. while SRF predicts a neuron has tuning for a particular angle through Fourier analysis, the consensus DeepTune images offer concrete and detailed visualization of these tunings. For example, for the bottom left neuron, both our method and SRF show an orientation tunings of about 70° and 120°.

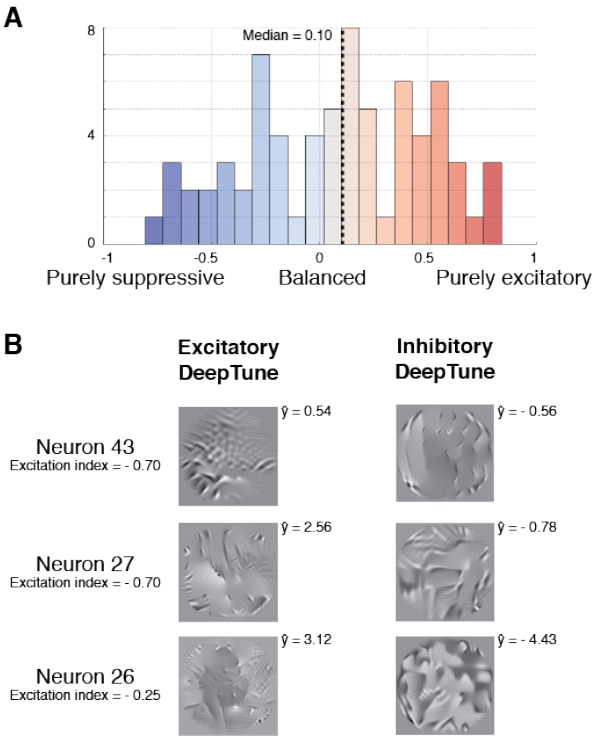


Figure 6.7. Neurons in the primate cortical area V4 exhibit suppressive tuning. **A.** Histogram of BWT excitation index for 71 V4 neurons. 41% of the neurons show strong suppressive tuning. The median of excitation index for V4 neurons is 0.10. **B.** The excitatory and inhibitory DeepTune images for three neurons identified as suppressive by the BWT model. The neuron excitation index and response of the model to each DeepTune image is illustrated in the same panel. The neurons with suppressive tuning have much clearer suppressive DeepTune images than those without. $\hat{y}$ is the predicted model response obtained by feeding the DeepTune image through AlexNet-Layer2 model.

Part V

Appendix

Appendix A

Techical proofs for the convergence of HMC

A.1 Proof of Lemmas 4, 5 and 6

In this appendix, we collect the proofs of Lemmas 4, and 5, as previously stated in Section 3.3.3, that are used in proving Theorem 3. Moreover, we provide the proof of auxiliary results related to HMC proposal that were used in the proof of Lemma 6.

A.1.1 Proof of Lemma 4

In order to prove Lemma 4, we begin by adapting the spectral profile technique [71] to the continuous state setting, and next we relate conductance profile with the spectral profile.

First, we briefly recall the notation from Section 2.2.1. Let $\Theta : \mathcal{X} \times \mathcal{B}(\mathcal{X}) \to \mathbb{R}_+$ denote the transition probability function for the Markov chain and let $\mathcal{T}$ be the corresponding transition operator, which maps a probability measure to another according to the transition probability Θ. Note that for a Markov chain satisfying the smooth chain assumption (3.12), if the distribution μ admits a density then the distribution $\mathcal{T}(\mu)$ would also admits a density. We use $\mathcal{T}_x$ as the shorthand for $\mathcal{T}(\delta_x)$, the transition distribution of the Markov chain at x.

Let $L_2(\pi^*)$ be the space of square integrable functions under function π^*. The *Dirichlet form* $\mathcal{E} : L_2(\pi^*) \times L_2(\pi^*) \to \mathbb{R}$ associated with the transition probability Θ is given by

$$\mathcal{E}(g,h) = \frac{1}{2}\int_{(x,y)\in\mathcal{X}^2} (g(x)-h(y))^2\,\Theta(x,dy)\pi^*(x)dx. \tag{A.1}$$

The expectation $\mathbb{E}_{\pi^*} : L_2(\pi^*) \to \mathbb{R}$ and the variance $\mathrm{Var}_{\pi^*} : L_2(\pi^*) \to \mathbb{R}$ with respect

to the density π^* are given by

$$\mathbb{E}_{\pi^*}(g) = \int_{x\in\mathcal{X}} g(x)\pi^*(x)dx \quad \text{and} \quad \mathrm{Var}_{\pi^*}(g) = \int_{x\in\mathcal{X}} \left(g(x) - \mathbb{E}_{\pi^*}(g)\right)^2 \pi^*(x)dx. \tag{A.2a}$$

Furthermore, for a pair of measurable sets $(S, \Omega) \subset \mathcal{X}^2$, the Ω-*restricted spectral gap for the set* S is defined as

$$\lambda_\Omega(S) = \inf_{g\in c_0^+(S\cap\Omega)} \frac{\mathcal{E}(g,g)}{\mathrm{Var}_{\pi^*}(g)}, \tag{A.3a}$$

$$\text{where } c_0^+(S\cap\Omega) = \{g \in L_2(\pi^*) \mid \mathrm{supp}(g) \subset S\cap\Omega, g \geq 0, g \neq \text{constant}\}. \tag{A.3b}$$

Finally, the Ω-*restricted spectral profile* Λ_Ω is defined as

$$\Lambda_\Omega(v) = \inf_{\Pi^*(S\cap\Omega)\in[0,v]} \lambda_\Omega(S\cap\Omega), \qquad \text{for all } v \in \big[0,\infty). \tag{A.4}$$

Note that we restrict the spectral profile to the set Ω. Taking Ω to be $\mathcal{X}$, our definition agrees with the standard definition definitions of the restricted spectral gap and spectral profile in the paper by Goel et al. [71] for finite state space Markov chains to continuous state space Markov chains.
We are now ready to state a mixing time bound using spectral profile.

Lemma 17. *Consider a reversible irreducible ζ-lazy Markov chain with stationary distribution Π^* satifying the smooth chain assumption (3.12). Given a ϖ-warm start μ_0, an error tolerance $\epsilon \in (0,1)$ and a set $\Omega \subset \mathcal{X}$ with $\Pi^*(\Omega) \geq 1 - \frac{\epsilon^2}{2\varpi^2}$, the L_2-mixing time is bounded as*

$$\tau_2(\epsilon;\mu_0) \leq \left\lceil \int_{4/\varpi}^{8/\epsilon^2} \frac{dv}{\zeta\cdot v\Lambda_\Omega(v)} \right\rceil, \tag{A.5}$$

where Λ_Ω denotes the Ω-restricted spectral profile (A.4) of the chain.

See Appendix A.1.1 for the proof.

In the next lemma, we state the relationship between the Ω-restricted spectral profile (A.4) of the Markov chain to its Ω-restricted conductance profile (3.10).

Lemma 18. *For a Markov chain with state space $\mathcal{X}$ and stationary distribution Π^*, given any measurable set $\Omega \subset \mathcal{X}$, its Ω-restricted spectral profile (A.4) and Ω-restricted conductance profile (3.10) are related as*

$$\Lambda_\Omega(v) \geq \begin{cases} \dfrac{\Phi_\Omega^2(v)}{2} & \text{for all } v \in \left[0, \frac{\Pi^*(\Omega)}{2}\right] \\ \dfrac{\Phi_\Omega^2(\Pi^*(\Omega)/2)}{4} & \text{for all } v \in \left(\frac{\Pi^*(\Omega)}{2}, \infty\right). \end{cases} \tag{A.6}$$

See Appendix A.1.1 for the proof.

Lemma 4 now follows from Lemmas 17 and 18 as well as the definition (3.11) of $\tilde{\Phi}_\Omega$.

Proof of Lemma 17

We need the following lemma, proved in for the case of finite state Markov chains in the paper [71], which lower bounds the Dirichlet form in terms of the spectral profile.

Lemma 19. *For any measurable set $\Omega \subset \mathcal{X}$, any non-constant function $g : \mathcal{X} \to \mathbb{R}_+$ such that $g \in L_2(\pi^*)$ and $supp(g) \subset \Omega$, we have*

$$\frac{\mathcal{E}(g,g)}{\mathrm{Var}_{\pi^*}(g)} \geq \frac{1}{2}\Lambda_\Omega\left(\frac{4\left(\mathbb{E}_{\pi^*}(g)\right)^2}{\mathrm{Var}_{\pi^*}(g)}\right). \tag{A.7}$$

The proof of Lemma 19 is a straightforward extension of Lemma 2.1 from Goel et al. [71], which deals with finite state spaces, to the continuous state Markov chain. See the end of Section A.1.1 for the proof.
We are now equipped to prove Lemma 17.

Proof of Lemma 17: We begin by introducing some notations. Recall that for any Markov chain satisfying the smooth chain assumption (3.12), given an initial distribution μ_0 that admits a density, the distribution of the chain at any step n also admits a density. As a result, we can define the ratio of the density of the Markov chain at the n-th iteration $h_{\mu_0,n} : \mathcal{X} \to \mathbb{R}$ with respect to the target density π^* via the following recursion

$$h_{\mu_0,0}(x) = \frac{\mu_0(x)}{\pi^*(x)} \quad \text{and} \quad h_{\mu_0,n+1}(x) = \frac{\mathcal{T}\left(\pi^* \cdot h_{\mu_0,n}\right)(x)}{\pi^*(x)},$$

where we have used the notation $\mathcal{T}(\mu)(x)$ to denote the density of the distribution $\mathcal{T}(\mu)$ at x. Note that

$$\mathbb{E}_{\pi^*}(h_{\mu_0,n}) = 1 \quad \text{and} \quad \mathbb{E}_{\pi^*}(h_{\mu_0,n} \cdot \mathbf{1}_\Omega) \leq 1 \quad \text{for all } n \geq 0, \tag{A.8}$$

where $\Omega \subset \mathcal{X}$ is a measurable set.

We also define the quantity $J(n) := \mathrm{Var}_{\pi^*}(h_{\mu_0,n})$ (we prove the existence of this variance below in Step (1)). Note that the L_2-distance between the distribution of the chain at step n and the target distribution is given by

$$d_{2,\pi^*}(\mathcal{T}^n(\mu_0), \Pi^*) = \left(\int_{x\in\mathbb{R}^d} \left(h_{\mu_0,n}(x) - 1\right)^2 \pi^*(x)dx\right)^{1/2} = \mathrm{Var}_{\pi^*}(h_{\mu_0,n}).$$

Consequently, to prove the ϵ-L_2 mixing time bound (A.5), it suffices to show that for any measurable set $\Omega \subset \mathcal{X}$, with $\Pi^*(\Omega) \geq 1 - \frac{\epsilon^2}{2\varpi^2}$, we have

$$J(n) \leq \epsilon^2 \quad \text{for } n \geq \left\lceil \int_{4/\varpi}^{8/\epsilon^2} \frac{dv}{\zeta \cdot v\Lambda_\Omega} \right\rceil \tag{A.9}$$

We now establish the claim (A.9) via a three step argument: (1) we prove the existence of the variance $J(n)$ for all $n \in \mathbb{N}$, (2) then we derive a recurrence relation for the difference $J(n+1) - J(n)$ in terms of Dirichlet forms that shows the J is a decreasing function, and (3) finally, using an extension of the variance J from natural indices to real numbers, we derive an explicit upper bound on the number of steps taken by the chain until J lies below the required threshold.

Step (1): Using the reversibility (2.4) of the chain, we find that

$$h_{\mu_0,n+1}(x)dx = \frac{\int_{y\in\mathcal{X}} \Theta(y,dx)h_{\mu_0,n}(y)\pi^*(y)dy}{\pi^*(x)} = \frac{\int_{y\in\mathcal{X}} \Theta(x,dy)h_{\mu_0,n}(y)\pi^*(x)dx}{\pi^*(x)}$$
$$= \int_{y\in\mathcal{X}} \Theta(x,dy)h_{\mu_0,n}(y)dx \qquad \text{(A.10)}$$

Applying an induction argument along with the relationship (A.10) and the initial condition $h_{\mu_0,0}(x) \leq \varpi$, we obtain that

$$h_{\mu_0,n}(x) \leq \varpi, \quad \text{for all } n \geq 0. \qquad \text{(A.11)}$$

As a result, the variances of the functions $h_{\mu_0,0}$ and $h_{\mu_0,n} \cdot \mathbf{1}_\Omega$ under the target density π^* are well-defined and

$$J(n) = \int_{\mathcal{X}} h^2_{\mu_0,n}(x)\pi^*(x)dx - 1 \qquad \text{(A.12)}$$

Step (2): We now bound the difference between consecutive variance terms. We have

$$J(n) - \mathrm{Var}_{\pi^*}(h_{\mu_0,n}\cdot\mathbf{1}_\Omega) = \mathrm{Var}_{\pi^*}(h_{\mu_0,n}) - \mathrm{Var}_{\pi^*}(h_{\mu_0,n}\cdot\mathbf{1}_\Omega)$$
$$= \int_{x\in\mathcal{X}\setminus\Omega} h^2_{\mu_0,n}(x)\pi^*(x)dx - \left(\int_{x\in\mathcal{X}} h_{\mu_0,n}(x)\pi^*(x)dx\right)^2$$
$$+ \left(\int_{x\in\Omega} h_{\mu_0,n}(x)\pi^*(x)dx\right)^2$$
$$\leq \varpi^2\left(1 - \Pi^*(\Omega)\right) \leq \frac{\epsilon^2}{2} =: B, \qquad \text{(A.13)}$$

where the last inequality follows from the fact that Ω satisfies $\Pi^*(\Omega) \geq 1 - \epsilon^2/(2\varpi^2)$. Also note the following bound on $J(0)$:

$$J(0) = \int_{x\in\mathcal{X}} \frac{\mu_0(x)^2}{\pi^*(x)}dx - 1 \leq \varpi \int_{x\in\mathcal{X}} \mu_0(x)dx - 1 \leq \varpi - 1. \qquad \text{(A.14)}$$

Define the two step transition kernel $\Theta \circ \Theta$ as

$$\Theta\circ\Theta(y,dz) = \int_{x\in\mathcal{X}} \Theta(y,dx)\Theta(x,dz).$$

We have

$$\begin{aligned} J(n+1) := \mathrm{Var}_{\pi^*}(h_{\mu_0,n+1}) &= \int_{x\in\mathcal{X}} h^2_{\mu_0,n+1}(x)\pi^*(x)dx - 1 \\ &\overset{(i)}{=} \int_{x\in\mathcal{X}}\int_{y\in\mathcal{X}} \Theta(y,dx)h_{\mu_0,n}(y)\pi^*(y)dy \int_{z\in\mathcal{X}} \Theta(x,dz)h_{\mu_0,n}(z) - 1 \\ &= \int_{y,z\in\mathcal{X}^2} \Theta\circ\Theta(y,dz)h_{\mu_0,n}(y)h_{\mu_0,n}(z)\pi^*(y)dy - 1, \end{aligned}$$

where step (i) follows from the relation (A.10). Using the above expression for $J(n+1)$ and the expression from equation (A.12) for $J(n)$, we find that

$$\begin{aligned} J(n+1) - J(n) &= \int_{\mathcal{X}^2} \Theta\circ\Theta(y,dz)h_{\mu_0,n}(y)h_{\mu_0,n}(z)\pi^*(y)dy - \int_{\mathcal{X}} h^2_{\mu_0,n}(x)\pi^*(x)dx, \\ &\overset{(a)}{=} -\mathcal{E}_{\Theta\circ\Theta}(h_{\mu_0,n}, h_{\mu_0,n}), \end{aligned} \tag{A.15}$$

where $\mathcal{E}_{\Theta\circ\Theta}$ is the Dirichlet form (A.1) with transition probability Θ being replaced by $\Theta\circ\Theta$. We come back to the proof of equality (a) at the end of this paragraph. Assuming it as given at the moment, we proceed further. Since the Markov chain is ζ-lazy, we can relate the two Dirichlet forms $\mathcal{E}_{\Theta\circ\Theta}$ and $\mathcal{E}_{\Theta}$ as follows: For any $y, z \in \mathcal{X}$ such that $y \neq z$, we have

$$\begin{aligned} \Theta\circ\Theta(y,dz) = \int_{x\in\mathcal{X}} \Theta(y,dx)\Theta(x,dz) &\geq \Theta(y,dy)\Theta(y,dz) + \Theta(y,dz)\Theta(z,dz) \\ &\geq 2\zeta\Theta(y,dz). \end{aligned} \tag{A.16}$$

We have

$$\begin{aligned} J(n+1) - J(n) = -\mathcal{E}_{\Theta\circ\Theta}(h_{\mu_0,n}, h_{\mu_0,n}) &\overset{(i)}{\leq} -2\zeta\mathcal{E}_{\Theta}(h_{\mu_0,n}, h_{\mu_0,n}) \\ &\overset{(ii)}{\leq} -2\zeta\mathcal{E}_{\Theta}(h_{\mu_0,n}\cdot\mathbf{1}_\Omega, h_{\mu_0,n}\cdot\mathbf{1}_\Omega) \\ &\overset{(iii)}{\leq} -\zeta\,\mathrm{Var}_{\pi^*}(h_{\mu_0,n}\cdot\mathbf{1}_\Omega)\Lambda_\Omega\left(\frac{4\left[\mathbb{E}_{\pi^*}(h_{\mu_0,n}\cdot\mathbf{1}_\Omega)\right]^2}{\mathrm{Var}_{\pi^*}(h_{\mu_0,n}\cdot\mathbf{1}_\Omega)}\right) \\ &\overset{(iv)}{\leq} -\zeta\cdot(J(n)-B)\,\Lambda_\Omega\left(\frac{4}{J(n)-B}\right). \end{aligned} \tag{A.17}$$

where step (i) follows from inequality (A.16), step (ii) follows from the fact that Dirichlet forms satisfy $\mathcal{E}_\Theta(h_{\mu_0,n}, h_{\mu_0,n}) \geq \mathcal{E}_\Theta(h_{\mu_0,n}\cdot\mathbf{1}_\Omega, h_{\mu_0,n}\cdot\mathbf{1}_\Omega)$, step (iii) follows from Lemma 19, and finally step (iv) follows from inequality (A.13) which implies that $\mathrm{Var}_{\pi^*}(h_{\mu_0,n}\cdot\mathbf{1}_\Omega) \geq J(n) - B$, and the fact that the spectral profile Λ_Ω is a non-increasing function.

Proof of equality (a) in equation (A.15): Since the distribution Π^* is stationary with respect to the kernel Θ, it is also stationary with respect to the two step kernel

$\Theta \circ \Theta$. We now prove a more general claim: For any transition kernel K which has stationary distribution Π^* and any measurable function h, the Dirichlet form $\mathcal{E}_K$, defined by replacing Θ with K in equation (A.1), we have

$$\mathcal{E}_K(h,h) = \int_{\mathcal{X}} h^2(x)\pi^*(x)dx - \int_{\mathcal{X}}\int_{\mathcal{X}} h(x)h(y)K(x,dy)\pi^*(x)dx. \tag{A.18}$$

Note that invoking this claim with $K = \Theta \circ \Theta$ and $h = h_{\mu_0,n}$ implies step (a) in equation (A.15). We now establish the claim (A.18). Expanding the square in the definition (A.1), we obtain that

$$\begin{aligned}\mathcal{E}_K(h,h) &= \frac{1}{2}\int_{\mathcal{X}}\int_{\mathcal{X}} h^2(x)K(x,dy)\pi^*(x)dx + \frac{1}{2}\int_{\mathcal{X}}\int_{\mathcal{X}} h^2(y)K(x,dy)\pi^*(x)dx \\ &\qquad - \int_{\mathcal{X}}\int_{\mathcal{X}} h(x)h(y)K(x,dy)\pi^*(x)dx \\ &\overset{(i)}{=} \frac{1}{2}\int_{\mathcal{X}} h^2(x)\pi^*(x)dx + \frac{1}{2}\int_{\mathcal{X}} h^2(x)\pi^*(x)dx - \int_{\mathcal{X}}\int_{\mathcal{X}} h(x)h(y)K(x,dy)\pi^*(x)dx,\end{aligned}$$

where equality (i) follows from the following facts: For the first term, we use the fact that $\int_{\mathcal{X}} K(x,dy) = 1$ since K is a transition kernel, and, for the second term we use the fact that $\int_{\mathcal{X}} K(x,dy)\pi^*(x)dx = \pi^*(y)dy$, since Π^* is the stationary distribution for the kernel K. The claim now follows.

Step (3): Consider the domain extension of the function J from $\mathbb{N}$ to the set of non-negative real numbers $\mathbb{R}_+$ by piecewise linear interpolation. We abuse notation and denote this extension also by J. The extended function J is continuous and is differentiable on the set $\mathbb{R}_+\backslash\mathbb{N}$. Let $n^* \in \mathbb{R}_+ \cup \{\infty\}$ denote the index such that $J(n^*) < B$. Since Λ_Ω is non-increasing and J is non-increasing, we have

$$J'(t) \leq -\zeta \cdot (J(t) - B)\,\Lambda_\Omega\left(\frac{4}{J(t)-B}\right) \quad \text{for all } t \in \mathbb{R}_+\backslash\mathbb{N} \text{ such that } t \leq n^*. \tag{A.19}$$

Moving the J terms on one side and integrating for $t \leq n^*$, we obtain

$$\int_{J(0)}^{J(t)} \frac{dJ}{(J-B)\cdot\Lambda_\Omega\left(\frac{4}{J-B}\right)} \leq -\zeta t.$$

Using the change of variable $v = 4/(J-B)$, we obtain

$$\zeta t \leq \int_{4/(J(0)-B)}^{4/(J(t)-B)} \frac{dv}{v\Lambda_\Omega(v)} \tag{A.20}$$

Furthermore, equation (A.20) implies that for $T \geq \frac{1}{\zeta}\int_{4/\varpi}^{8/\epsilon^2} \frac{dv}{v\Lambda_\Omega(v)}$, we have

$$\int_{4/\varpi}^{8/\epsilon^2} \frac{dv}{v\Lambda_\Omega(v)} \leq \int_{4/(J(0)-B)}^{4/(J(T)-B)} \frac{dv}{v\Lambda_\Omega(v)}.$$

The bound (A.14) and the fact that $B = \epsilon^2/2$ imply that $4/(J(0) - B) > 4/\varpi$. Using this observation and the fact that $0 \leq \Lambda_\Omega(v) < \infty$ for $v \geq 4/\varpi$, we conclude that

$$J(T) \leq B = \frac{\epsilon^2}{2} \text{ or } \frac{4}{J(T) - B} \geq \frac{8}{\epsilon^2} \quad \text{for } T \geq \frac{1}{\zeta} \int_{4/\varpi}^{8/\epsilon^2} \frac{dv}{v\Lambda(v)},$$

which implies the claimed bound (A.9).

Finally, we turn to the proof of Lemma 19.

Proof of Lemma 19: Fix a non-constant function $g : \mathcal{X} \to \mathbb{R}_+$ such that $g \in L_2(\pi^*)$ and $\text{supp}(g) \subset \Omega$. Note that for any constant $c \in \mathbb{R}$, we have

$$\begin{aligned}
\mathcal{E}(g, g) &= \frac{1}{2} \int_{(x,y)\in\mathcal{X}^2} (g(x) - g(y))^2 \, \Theta(x, dy)\Pi^*(x)dx \\
&= \frac{1}{2} \int_{(x,y)\in\Omega^2} (g(x) - g(y))^2 \, \Theta(x, dy)\Pi^*(x)dx \\
&= \frac{1}{2} \int_{(x,y)\in\Omega^2} ((g(x) - c) - (g(y) - c))^2 \, \Theta(x, dy)\Pi^*(x)dx \\
&= \mathcal{E}\left((g - c) \cdot \mathbf{1}_\Omega, (g - c) \cdot \mathbf{1}_\Omega\right).
\end{aligned}$$

Consequently, we obtain that

$$\begin{aligned}
\mathcal{E}(g, g) = \mathcal{E}\left((g - c) \cdot \mathbf{1}_\Omega, (g - c) \cdot \mathbf{1}_\Omega\right) &\geq \mathcal{E}\left((g - c)_+ \cdot \mathbf{1}_\Omega, (g - c)_+ \cdot \mathbf{1}_\Omega\right) \\
&\overset{(i)}{\geq} \text{Var}_{\pi^*}\left((g - c)_+ \cdot \mathbf{1}_\Omega\right) \inf_{f \in c_0^+(\{g>c\}\cap\Omega)} \frac{\mathcal{E}(f, f)}{\text{Var}_{\pi^*}(f)} \\
&\overset{(ii)}{\geq} \text{Var}_{\pi^*}\left((g - c)_+ \cdot \mathbf{1}_\Omega\right) \cdot \Lambda_\Omega(\Pi^*(\{g > c\} \cap \Omega)).
\end{aligned} \tag{A.21}$$

Here $(x)_+ = \max\{0, x\}$ denotes the positive part of x. Inequality (i) follows from the infimum and inequality (ii) follows from the definition (A.4) of Ω-restricted spectral profile. Additionally, we have

$$\begin{aligned}
\text{Var}_{\pi^*}\left((g - c)_+ \cdot \mathbf{1}_\Omega\right) &= \mathbb{E}_{\pi^*}\left((g - c)_+ \cdot \mathbf{1}_\Omega\right)^2 - \left[\mathbb{E}_{\pi^*}\left((g - c)_+ \cdot \mathbf{1}_\Omega\right)\right]^2 \\
&\overset{(i)}{\geq} \mathbb{E}_{\pi^*}(g)^2 - 2(c\Pi^*(\Omega)) \cdot \mathbb{E}_{\pi^*}(g) - \left[\mathbb{E}_{\pi^*}(g)\right]^2 \\
&\geq \text{Var}_{\pi^*}(g) - 2c\mathbb{E}_{\pi^*}(g),
\end{aligned} \tag{A.22}$$

where inequality (i) follows from the fact that

$$(a - b)_+^2 \geq a^2 - 2ab \quad \text{and} \quad (a - b)_+ \leq a, \quad \text{for scalars } a, b \geq 0.$$

Setting $c = \mathrm{Var}_{\pi^*}(g)/4\mathbb{E}_{\pi^*}(g)$, we obtain from equation (A.22) that

$$\mathrm{Var}_{\pi^*}\left((g-c)_+ \mathbf{1}_\Omega\right) \geq \frac{1}{2}\mathrm{Var}_{\pi^*}(g) \tag{A.23}$$

Furthermore for any $c > 0$, applying Markov's inequality for the non-negative function $g \cdot \mathbf{1}_\Omega$, we also have $\Pi^*(\{g > c\} \cap \Omega) \leq \Pi^*(\{g > c\}) \leq [\mathbb{E}_{\pi^*}(g)]/c$. Combing equation (A.21) and (A.23), together with the fact that Λ_Ω is non-increasing, we obtain

$$\mathcal{E}(g,g) \geq \frac{1}{2}\mathrm{Var}_{\pi^*}(g) \cdot \Lambda_\Omega\left(\frac{4\left(\mathbb{E}_{\pi^*}(g)\right)^2}{\mathrm{Var}_{\pi^*}(g)}\right),$$

as claimed in the lemma.

Proof of Lemma 18

The proof of the Lemma 18 follows along the lines of Lemma 2.4 in the papre [71], except that we have to deal with continuous-state transition probability. This technical challenge is the main reason for introducing the restricted conductance profile. At a high level, our argument is based on reducing the problem on general functions to a problem on indicator functions, and then using the definition of the conductance. Similar ideas have appeared in the proof of the Cheeger's inequality [33] and the modified log-Sobolev constants [81].

We split the proof of Lemma 18 in two cases based on whether $v \in [\frac{4}{\varpi}, \frac{\Pi^*(\Omega)}{2}]$, referred to as Case 1, or $v \geq \frac{\Pi^*(\Omega)}{2}$, referred to as Case 2.

Case 1: First we consider the case when $v \in [\frac{4}{\varpi}, \frac{\Pi^*(\Omega)}{2}]$. First, we define $D^+ : L_2(\pi^*) \to L_2(\pi^*)$ as

$$D^+(g)(x) = \int_{y\in\mathcal{X}} (g(x)-g(y))_+ \Theta(x,dy) \text{ and } D^-(g)(x) = \int_{y\in\mathcal{X}} (g(x)-g(y))_- \Theta(x,dy),$$

where $(x)_+ = \max\{0,x\}$ and (resp. $(\cdot)_-$) denote the positive and negative part of x respectively. We note that D^+ and D^- satisfy the following co-area formula:

$$\mathbb{E}_{\pi^*} D^+(g) = \int_{-\infty}^{+\infty} \mathbb{E}_{\pi^*} D^+ \mathbf{1}_{g>t} dt. \tag{A.24a}$$

See Lemma 1 in the paper [81] or Lemma 2.4 in the paper [71] for a proof of the equality (A.24a). Moreover, given any measurable set $A \subset \mathcal{X}$, scalar t, and function $g \in c_0^+(A \cap \Omega)$, we note that the term $\mathbb{E}_{\pi^*} D^+(\mathbf{1}_{g>t})(x)$ is equal to the flow ϕ (defined in equation (3.9)) of the level set $G_t = \{x \in \Omega \mid g(x) > t\}$:

$$\mathbb{E}_{\pi^*} D^+(\mathbf{1}_{g>t}) = \int_{x\in G_t} \Theta(x, G_t^c)\pi^*(x)dx = \phi(G_t). \tag{A.24b}$$

Since $G_t \subset \Omega$, we have

$$\phi(G_t) \geq \Pi^*(G_t) \cdot \inf_{0 \leq \Pi^*(S \cap \Omega) \leq \Pi^*(A \cap \Omega)} \frac{\phi(S)}{\Pi^*(S \cap \Omega)}. \tag{A.24c}$$

Combining the previous three equations, we find that[1]

$$\begin{aligned}\mathbb{E}_{\pi^*} D^+(g) = \int_{-\infty}^{+\infty} \mathbb{E}_{\pi^*} D^+ \mathbf{1}_{g>t} dt &\geq \int_{-\infty}^{+\infty} \Pi^*(G_t) dt \cdot \inf_{0 \leq \Pi^*(S \cap \Omega) \leq \Pi^*(A \cap \Omega)} \frac{\phi(S)}{\Pi^*(S \cap \Omega)} \\ &= \mathbb{E}_{\pi^*}(g) \cdot \Phi_\Omega(\Pi^*(A \cap \Omega)).\end{aligned}$$

In a similar fashion, we also obtain that

$$\mathbb{E}_{\pi^*} D^-(g) \geq \mathbb{E}_{\pi^*}(g) \cdot \Phi_\Omega(\Pi^*(A \cap \Omega)).$$

Combining these two bounds, we find that

$$\int_{\mathcal{X}} \int_{\mathcal{X}} |g(x) - g(y)| \, \Theta(x, dy) \pi^*(x) dx = \mathbb{E}_{\pi^*} D^+(g) + \mathbb{E}_{\pi^*} D^-(g) \geq 2\mathbb{E}_{\pi^*}(g) \cdot \Phi_\Omega(\Pi^*(A \cap \Omega)).$$

Applying this inequality with the function g^2, we have

$$\begin{aligned}
&2\mathbb{E}_{\pi^*}(g^2) \cdot \Phi_\Omega(\Pi^*(A \cap \Omega)) \\
&\leq \int_{\mathcal{X}} \int_{\mathcal{X}} \left| g^2(x) - g^2(y) \right| \Theta(x, dy) \pi^*(x) dx \\
&= \int_{\mathcal{X}} \int_{\mathcal{X}} |g(x) - g(y)| \, |g(x) + g(y)| \, \Theta(x, dy) \pi^*(x) dx \\
&\overset{(i)}{\leq} \left(\int_{\mathcal{X}} \int_{\mathcal{X}} |g(x) - g(y)|^2 \, \Theta(x, dy) \pi^*(x) dx \right)^{1/2} \cdot \left(\int_{\mathcal{X}} \int_{\mathcal{X}} |g(x) + g(y)|^2 \, \Theta(x, dy) \pi^*(x) dx \right)^{1/2} \\
&\overset{(ii)}{\leq} (2\mathcal{E}(g, g))^{1/2} \cdot \left(\int_{\mathcal{X}} \int_{\mathcal{X}} 2 \left(g(x)^2 + g(y)^2 \right) \Theta(x, dy) \pi^*(x) dx \right)^{1/2} \\
&= (2\mathcal{E}(g, g))^{1/2} \left(4\mathbb{E}_{\pi^*}(g^2) \right)^{1/2}.
\end{aligned}$$

Rearranging the last equation, we obtain that

$$\frac{\mathcal{E}(g, g)}{\mathbb{E}_{\pi^*}(g^2)} \geq \frac{\Phi_\Omega^2(\Pi^*(A \cap \Omega))}{2}. \tag{A.25}$$

In the above sequence of steps, inequality (i) follows from the Cauchy-Schwarz inequality, and inequality (ii) from the definition (A.1) and the fact that $(a+b)^2 \leq 2(a^2+b^2)$. Taking infimum over $g \in c_0^+(A \cap \Omega)$ in equation (A.25), we obtain

$$\lambda_\Omega(A) = \inf_{g \in c_0^+(A \cap \Omega)} \frac{\mathcal{E}(g, g)}{\mathrm{Var}_{\pi^*}(g)} \geq \inf_{g \in c_0^+(A \cap \Omega)} \frac{\mathcal{E}(g, g)}{\mathbb{E}_{\pi^*}(g^2)} \geq \frac{\Phi_\Omega^2(\Pi^*(A \cap \Omega))}{2},$$

[1] Note that this step demonstrates that the continuous state-space treatment is different from the discrete state-space one in Lemma 2.4 of Goel et al. [71].

where the first inequality follows from the fact that $\mathbb{E}_{\pi^*}(g^2) \geq \mathrm{Var}_{\pi^*}(g)$. Given $v \in [0, \frac{\Pi^*(\Omega)}{2}]$, taking infimum over $\Pi^*(A \cap \Omega) \leq v$ on both sides, we conclude the claimed bound for this case:

$$\Lambda_\Omega(v) = \inf_{\Pi^*(A\cap\Omega)\in[0,v]} \lambda_\Omega(A) \geq \inf_{\Pi^*(A\cap\Omega)\in[0,v]} \frac{\Phi^2_\Omega(\Pi^*(A\cap\Omega))}{2} = \frac{\Phi^2_\Omega(v)}{2},$$

where the last equality follows from the fact that the conductance profile Φ_Ω defined in equation (3.10) is non-increasing over its domain $[0, \frac{\Pi^*(\Omega)}{2}]$.

Case 2: Next, we consider the case when $v \geq \frac{\Pi^*(\Omega)}{2}$. We claim that

$$\Lambda_\Omega(v) \overset{(i)}{\geq} \Lambda_\Omega(\Pi^*(\Omega)) \overset{(ii)}{\geq} \frac{\Lambda_\Omega(\Pi^*(\Omega)/2)}{2} \overset{(iii)}{\geq} \frac{\Phi_\Omega(\Pi^*(\Omega)/2)^2}{4}, \tag{A.26}$$

where step (i) follows from the fact that the spectral profile Λ is a non-increasing function, and step (iii) from the result of Case 1. Note that the bound from Lemma 18 for this case follows from the bound above. It remains to establish inequality (ii), which we now prove.

Note that given the definition (A.4), it suffices to establish that

$$\frac{\mathcal{E}(g,g)}{\mathrm{Var}_{\pi^*}(g)} \geq \frac{\Lambda_\Omega(\Pi^*(\Omega)/2)}{2} \quad \text{for all functions } g \in c_0^+(\Omega). \tag{A.27}$$

Consider any fixed $g \in c_0^+(\Omega)$ and let $\nu \in \mathbb{R}$ be such that

$$\Pi^*(\{g > \nu\} \cap \Omega) = \Pi^*(\{g < \nu\} \cap \Omega) = \frac{\Pi^*(\Omega)}{2}.$$

Using the same argument as in the proof of Lemma 19, we have

$$\begin{aligned}\mathcal{E}(g,g) &= \mathcal{E}((g-\nu)\cdot\mathbf{1}_\Omega, (g-\nu)\cdot\mathbf{1}_\Omega) \\ &\geq \mathcal{E}((g-\nu)_+\cdot\mathbf{1}_\Omega, (g-\nu)_+\cdot\mathbf{1}_\Omega) + \mathcal{E}((g-\nu)_-\cdot\mathbf{1}_\Omega, (g-\nu)_-\cdot\mathbf{1}_\Omega).\end{aligned} \tag{A.28}$$

We have

$$\mathcal{E}((g-\nu)_+\cdot\mathbf{1}_\Omega, (g-\nu)_+\cdot\mathbf{1}_\Omega) \geq \mathbb{E}_{\pi^*}\left((g-\nu)_+^2\cdot\mathbf{1}_\Omega\right) \cdot \inf_{f\in c_0^+(\{g>\nu\}\cap\Omega)} \frac{\mathcal{E}(f,f)}{\mathbb{E}_{\pi^*} f^2}, \tag{A.29}$$

and similarly

$$\mathcal{E}((g-\nu)_-\cdot\mathbf{1}_\Omega, (g-\nu)_-\cdot\mathbf{1}_\Omega) \geq \mathbb{E}_{\pi^*}\left((g-\nu)_-^2\cdot\mathbf{1}_\Omega\right) \cdot \inf_{f\in c_0^+(\{g<\nu\}\cap\Omega)} \frac{\mathcal{E}(f,f)}{\mathbb{E}_{\pi^*} f^2}. \tag{A.30}$$

For $f \in c_0^+(\{g > \nu\} \cap \Omega)$, using Cauchy-Schwarz inequality, we have

$$\mathbb{E}_{\pi^*} f^2 = \int_{x\in\{g>\nu\}\cap\Omega} f(x)^2 \Pi^*(x)dx \geq \frac{\left(\int_{x\in\{g>\nu\}\cap\Omega} |f(x)|\,\Pi^*(x)dx\right)^2}{\Pi^*(\{g>\nu\}\cap\Omega)}$$

Using this bound and noting the ν is chosen such that $\Pi^*(\{g > \nu\} \cap \Omega) = \Pi^*(\Omega)/2$, for $f \in c_0^+(\{g > \nu\} \cap \Omega)$, we have

$$\mathrm{Var}_{\pi^*}(f) = \mathbb{E}_{\pi^*} f^2 - (\mathbb{E}_{\pi^*} f)^2 \geq \mathbb{E}_{\pi^*} f^2 \cdot \left(1 - \frac{\Pi^*(\Omega)}{2}\right). \tag{A.31}$$

Putting the equations (A.28), (A.29), (A.30) and (A.31) together, we obtain

$$\begin{aligned}\mathcal{E}(g,g) &\geq \mathbb{E}_{\pi^*}\left((g-\nu)^2 \cdot \mathbf{1}_\Omega\right) \cdot \left(1 - \frac{\Pi^*(\Omega)}{2}\right) \cdot \inf_{\Pi^*(S) \in [0, \frac{\Pi^*(\Omega)}{2}]} \inf_{f \in c_0^+(S \cap \Omega)} \frac{\mathcal{E}(f,f)}{\mathrm{Var}_{\pi^*}(f)} \\ &= \mathrm{Var}_{\pi^*}(g) \cdot \frac{1}{2} \cdot \Lambda_\Omega(\Pi^*(\Omega)/2).\end{aligned}$$

which implies the claim (A.27) and we are done.

A.1.2 Proof of Lemma 5

The proof of this lemma is similar to the conductance based proof for continuous Markov chains (see, e.g., Lemma 2 in our past work [58]). In addition to it, we have to deal with the case when target distribution satisfies the logarithmic isoperimetric inequality.

For any set A_1 such that $\Pi^*(A_1 \cap \Omega) \leq \frac{\Pi^*(\Omega)}{2}$, with its complement denoted by $A_2 = \mathcal{X} \backslash A_1$, we have $\Pi^*(A_2 \cap \Omega) \geq \frac{\Pi^*(\Omega)}{2} \geq \Pi^*(A_1 \cap \Omega)$, since $\Pi^*(A_1 \cap \Omega) + \Pi^*(A_2 \cap \Omega) = \Pi^*(\Omega)$. We claim that

$$\int_{x \in A_1} \Theta(x, A_2) \pi^*(x) dx \geq \Pi^*(A_1 \cap \Omega) \cdot \frac{\omega}{4} \cdot \min\left\{1, \frac{\Delta}{16 \psi_{\mathfrak{a}}} \cdot \log^{\mathfrak{a}}\left(1 + \frac{1}{\Pi^*(A_1 \cap \Omega)}\right)\right\}. \tag{A.32}$$

Note that the claim (3.16) of Lemma 5 can be directly obtained from the claim (A.32), by dividing both sides by $\Pi^*(A_1 \cap \Omega)$, taking infimum with respect to A_1 such $\Pi^*(A_1 \cap \Omega) \in (0, v]$ and noting that $\inf_{t \in (0,v]} \log^{\frac{1}{2}}(1 + 1/t) = \log^{\frac{1}{2}}(1 + 1/v)$.

We now prove the claim (A.32).

Define the following sets,

$$A_1' := \left\{x \in A_1 \cap \Omega \mid \Theta(x, A_2) < \frac{\omega}{2}\right\}, \quad A_2' := \left\{x \in A_2 \cap \Omega \mid \Theta(x, A_1) < \frac{\omega}{2}\right\}, \tag{A.33}$$

along with the complement $A_3' := \Omega \setminus (A_1' \cup A_2')$. Note that $A_i' \subset \Omega$ for $i = 1, 2, 3$. We split the proof into two distinct cases:

- Case 1: $\Pi^*(A_1') \leq \Pi^*(A_1 \cap \Omega)/2$ or $\Pi^*(A_2') \leq \Pi^*(A_2 \cap \Omega)/2$.
- Case 2: $\Pi^*(A_1') > \Pi^*(A_1 \cap \Omega)/2$ and $\Pi^*(A_2') > \Pi^*(A_2 \cap \Omega)/2$.

Note that these cases are mutually exclusive and exhaustive. We now consider these cases one by one.

Case 1: If we have $\Pi^*(A_1') \leq \Pi^*(A_1 \cap \Omega)/2$, then

$$\Pi^*(A_1 \cap \Omega \setminus A_1') \geq \Pi^*(A_1 \cap \Omega)/2. \tag{A.34}$$

We have

$$\int_{x \in A_1} \Theta(x, A_2)\pi^*(x)dx \geq \int_{x \in A_1 \cap \Omega \setminus A_1'} \Theta(x, A_2)\pi^*(x)dx \overset{(i)}{\geq} \frac{\omega}{2} \int_{x \in A_1 \cap \Omega \setminus A_1'} \pi^*(x)dx \overset{(ii)}{\geq} \frac{\omega}{4}\Pi^*(A_1 \cap \Omega),$$

where inequality (i) follows from the definition of the set A_1' in equation (A.33) and inequality (ii) follows from equation (A.34). For the case $\Pi^*(A_2') \leq \Pi^*(A_2 \cap \Omega)/2$, we use a similar argument with the role of A_1 and A_2 exchanged to obtain

$$\int_{x \in A_1} \Theta(x, A_2)\pi^*(x)dx = \int_{x \in A_2} \Theta(x, A_1)\pi^*(x)dx \geq \frac{\omega}{4}\Pi^*(A_2 \cap \Omega).$$

Putting the pieces together for this case, we have established that

$$\int_{x \in A_1} \Theta(x, A_2)\pi^*(x)dx \geq \frac{\omega}{4} \min\left\{\Pi^*(A_1 \cap \Omega), \Pi^*(A_2 \cap \Omega)\right\} = \frac{\omega}{4}\Pi^*(A_1 \cap \Omega). \tag{A.35}$$

Case 2: We have $\Pi^*(A_1') > \Pi^*(A_1 \cap \Omega)/2$ and $\Pi^*(A_2') > \Pi^*(A_2 \cap \Omega)/2$. We first show that in this case the sets A_1' and A_2' are far away, and then we invoke the logarithmic isoperimetry inequality from Lemma 21.

For any two vectors $u \in A_1'$ and $v \in A_2'$, we have

$$d_{\mathrm{TV}}\left(\mathcal{T}_u, \mathcal{T}_v\right) \geq \Theta(u, A_1) - \Theta(v, A_1) = 1 - \Theta(u, A_2) - \Theta(v, A_1) > 1 - \omega.$$

Consequently, the assumption of the lemma implies that

$$d(A_1', A_2') \geq \Delta. \tag{A.36}$$

Using the fact that under the stationary distribution, the flow from A_1 to A_2 is equal to that from A_2 to A_1, we obtain

$$\begin{aligned}
&\int_{x \in A_1} \Theta(x, A_2)\pi^*(x)dx \\
&= \frac{1}{2}\left(\int_{x \in A_1} \Theta(x, A_2)\pi^*(x)dx + \int_{x \in A_2} \Theta(x, A_1)\pi^*(x)dx\right) \\
&\geq \frac{1}{4}\left(\int_{x \in A_1 \cap \Omega \setminus A_1'} \Theta(x, A_2)\pi^*(x)dx + \int_{x \in x \in A_2 \cap \Omega \setminus A_2'} \Theta(x, A_1)\pi^*(x)dx\right) \\
&\geq \frac{\omega}{8}\Pi^*(\Omega \setminus (A_1' \cup A_2')),
\end{aligned} \tag{A.37}$$

where the last inequality follows from the definition of the set A_1' in equation (A.33). Note that the sets A_1', A_2' and $\mathcal{X} \setminus (A_1' \cup A_2')$ partition $\mathcal{X}$. Using the condition (3.6d) with the Ω-restricted distribution Π^*_Ω with density π^*_Ω defined as

$$\pi^*_\Omega(x) = \frac{\pi^*(x)\mathbf{1}_\Omega(x)}{\Pi^*(\Omega)},$$

we obtain

$$\begin{aligned}
&\Pi^*(\Omega \setminus (A_1' \cap A_2')) \\
&= \Pi^*(\Omega) \cdot \Pi^*_\Omega(\mathcal{X} \setminus (A_1' \cap A_2')) \\
&\overset{(i)}{\geq} \Pi^*(\Omega) \cdot \frac{d(A_1', A_2')}{2\psi_{\mathfrak{a}}} \cdot \min\left\{\Pi^*_\Omega(A_1'), \Pi^*_\Omega(A_2')\right\} \cdot \log^{\mathfrak{a}}\left(1 + \frac{1}{\min\left\{\Pi^*_\Omega(A_1'), \Pi^*_\Omega(A_2')\right\}}\right) \\
&\overset{(ii)}{\geq} \Pi^*(\Omega) \cdot \frac{\Delta}{4\psi_{\mathfrak{a}}} \min\left\{\Pi^*(A_1 \cap \Omega), \Pi^*(A_2 \cap \Omega)\right\} \cdot \log^{\mathfrak{a}}\left(1 + \frac{2}{\min\left\{\Pi^*(A_1 \cap \Omega), \Pi^*(A_2 \cap \Omega)\right\}}\right) \\
&\geq \frac{1}{2} \cdot \frac{\Delta}{4\psi_{\mathfrak{a}}} \cdot \Pi^*(A_1 \cap \Omega) \cdot \log^{\mathfrak{a}}\left(1 + \frac{1}{\Pi^*(A_1 \cap \Omega)}\right),
\end{aligned} \tag{A.38}$$

where step (i) follows from the assumption (3.6d), step (ii) from the bound (A.36) and the facts that $\Pi^*_\Omega(A_i') \geq \Pi^*(A_i') \geq \frac{1}{2}\Pi^*(A_i \cap \Omega)$ and that the map $x \mapsto x \log^{\mathfrak{a}}(1 + 1/x)$ is an increasing function for either $\mathfrak{a} = \frac{1}{2}$ or $\mathfrak{a} = 0$. Putting the pieces (A.37) and (A.38) together, we conclude that

$$\int_{x \in A_1} \Theta(x, A_2)\pi^*(x)dx \geq \frac{\omega}{16} \cdot \frac{\Delta}{4\psi_{\mathfrak{a}}} \cdot \Pi^*(A_1 \cap \Omega) \cdot \log^{\mathfrak{a}}\left(1 + \frac{1}{\Pi^*(A_1 \cap \Omega)}\right). \tag{A.39}$$

Finally, the claim (A.32) follows from combining the two bounds (A.35) and (A.39) from the two separate cases.

A.1.3 Proofs related to Lemma 6

We now present the proof of the intermediate results related to the HMC chain that were used in the proof of Lemma 6, namely, Lemmas 7, 8, 9 and 10. For simplicity, we adopt following the tensor notation.

Notations for tensor: Let $\mathcal{T} \in \mathbb{R}^{d \times d \times d}$ be a third order tensor. Let $U \in \mathbb{R}^{d \times d_1}$, $V \in \mathbb{R}^{d \times d_2}$, and $W \in \mathbb{R}^{d \times d_3}$ be three matrices. Then the multi-linear form applied on (U, V, W) is a tensor in $\mathbb{R}^{d_1 \times d_2 \times d_3}$:

$$[\mathcal{T}(U, V, W)]_{p,q,r} = \sum_{i,j,k \in [d]} \mathcal{T}_{ijk} U_{ip} V_{jq} W_{kr}.$$

In particular, for the vectors $u, v, w \in \mathbb{R}^d$, the quantity $\mathcal{T}(u, v, w)$ is a real number that depends linearly on u, v, w (tensor analogue of the quantity $u^\top M v$ in the context of matrices and vector). Moreover, the term $\mathcal{T}(u, v, \mathbb{I}_d)$ denotes a vector in $\mathbb{R}^d$ (tensor analogue of the quantity Mv in the context of matrices and vector). Finally, the term $\mathcal{T}(u, \mathbb{I}_d, \mathbb{I}_d)$ represents a matrix in $\mathbb{R}^{d \times d}$.

Proof of Lemma 7

We will prove an equivalent statement: for $K^2\eta^2 \leq \frac{1}{4L}$, there is a matrix $Q(x,y) \in \mathbb{R}^{d\times d}$ with $\|Q\|_2 \leq \frac{1}{8}$ such that

$$\mathbf{J}_x F(x,y) = K\eta\left(\mathbb{I}_d - Q(x,y)\right), \quad \text{for all } x, y \in \mathcal{X}. \tag{A.40}$$

Recall from equation (3.26b) that the intermediate iterate q_k is defined recursively as

$$q_k = F_k(p_0,q_0) = q_0 + k\eta p_0 - \frac{k\eta^2}{2}\nabla f(q_0) - \eta^2 \sum_{j=1}^{k-1}(k-j)\nabla f(q_j) \quad \text{for} \quad 1 \leq k \leq K.$$

Taking partial derivative with respective to the first variable, we obtain

$$\frac{\partial}{\partial p_0} q_k = \mathbf{J}_{p_0} F_k(p_0,q_0) = k\eta\mathbb{I}_d - \eta^2 \sum_{j=1}^{k-1}(k-j)\nabla^2 f_{q_j}\mathbf{J}_{p_0}F_j(p_0,q_0), \tag{A.41}$$

where $\nabla^2 f_{q_j}$ is the Hessian of f at q_j. We claim that for $1 \leq k \leq K$, there is a matrix $Q_k \in \mathbb{R}^{d\times d}$ with $\|Q_k\|_2 \leq \frac{1}{8}$ such that

$$\mathbf{J}_{p_0} F_k(p_0,q_0) = k\eta\left(\mathbb{I}_d - Q_k\right). \tag{A.42}$$

Note that substituting $k = K$ in this claim yields the result of the lemma. We now prove the claim (A.42) using strong induction.

Base case ($k = 1, 2$): For the base case $k = 1, 2$, using equation (A.41), we have

$$\mathbf{J}_{p_0} F_1(p_0,q_0) = \eta\mathbb{I}_d, \quad \text{and}$$
$$\mathbf{J}_{p_0} F_2(p_0,q_0) = 2\eta\mathbb{I}_d - \eta^2\nabla^2 f_{q_1}\mathbf{J}_{p_0}F_1(p_0,q_0) = 2\eta\left(\mathbb{I}_d - \frac{\eta^2}{2}\nabla^2 f_{q_1}\right).$$

Combining the inequality $\left\|\nabla^2 f_{q_1}\right\|_2 \leq L$ from smoothness assumption and the assumed stepsize bound $\eta^2 \leq \frac{1}{4L}$ yields

$$\left\|\frac{\eta^2}{2}\nabla^2 f_{q_1}\right\|_2 \leq \frac{1}{8}.$$

The statement in equation (A.42) is verified for $k = 1, 2$.

Inductive step: Assuming that the hypothesis holds for all iterations up to k, we now establish it for iteration $k+1$. We have

$$\begin{aligned}
\mathbf{J}_{p_0} F_{k+1}(p_0,q_0) &= (k+1)\eta\mathbb{I}_d - \eta^2\sum_{j=1}^{k}(k+1-j)\nabla^2 f_{q_j}\mathbf{J}_{p_0}F_j(p_0,q_0) \\
&\overset{(i)}{=} (k+1)\eta\mathbb{I}_d - \eta^2\sum_{j=1}^{k}(k+1-j)\nabla^2 f_{q_j}\cdot j\eta\left(\mathbb{I}_d - Q_j\right) \\
&= (k+1)\eta(\mathbb{I}_d - Q_{k+1}),
\end{aligned}$$

where $Q_{k+1} = \frac{\eta^2}{k+1}\sum_{j=1}^{k}(k+1-j)j\nabla^2 f_{q_j}(\mathbb{I}_d - Q_j)$. Equality (i) follows from the hypothesis of the induction. Finally, we verify that the spectral norm of Q_{k+1} is bounded by $\frac{1}{8}$,

$$\begin{aligned}
\|\!|Q_{k+1}|\!\|_2 &\leq \frac{1}{k+1}\sum_{j=1}^{k}\left\|\!\left|\eta^2(k+1-j)j\nabla^2 f_{q_j}\right|\!\right\|_2 \|\!|\mathbb{I}_d - Q_j|\!\|_2 \\
&\overset{(i)}{\leq} \frac{1}{k+1}\sum_{j=1}^{k}\left\|\!\left|\eta^2\frac{K^2}{4}\nabla^2 f_{q_j}\right|\!\right\|_2 \|\!|\mathbb{I}_d - Q_j|\!\|_2 \\
&\overset{(ii)}{\leq} \frac{1}{k+1}\sum_{j=1}^{k}\frac{1}{16}\left(1+\frac{1}{8}\right) \\
&\leq \frac{1}{8}.
\end{aligned}$$

Inequality (i) follows from the inequality $(k+1-j)j \leq \left(\frac{k+1-j+j}{2}\right)^2 \leq \frac{K^2}{4}$. Inequalilty (ii) follows from the assumption $K^2\eta^2 \leq \frac{1}{4L}$ and the hypothesis $\|\!|Q_j|\!\|_2 \leq \frac{1}{8}$. This completes the induction.

Proof of Lemma 8

Recall that the backward mapping G is defined implicitly as

$$x = y + K\eta G(x,y) - \frac{K\eta^2}{2}\nabla f(y) - \eta^2\sum_{k=1}^{K-1}(K-k)\nabla f\left(F_k(G(x,y),y)\right). \tag{A.43}$$

First we check the derivatives of $F_k(G(x,y),y)$. Since $F_k(G(x,y),y)$ satisfies

$$F_k(G(x,y),y) = y + k\eta G(x,y) - \frac{k\eta^2}{2}\nabla f(y) - \eta^2\sum_{j=1}^{k-1}(k-j)\nabla f(F_j(G(x,y),y)),$$

taking derivative with respect to y, we obtain

$$\begin{aligned}
\frac{\partial}{\partial y}F_k(G(x,y),y) = \mathbb{I}_d + k\eta\mathbf{J}_y G(x,y) - \frac{k\eta^2}{2}\nabla^2 f(y) \\
- \eta^2\sum_{j=1}^{k-1}(k-j)\nabla^2 f(F_j(G(x,y),y))\frac{\partial}{\partial y}F_j(G(x,y),y).
\end{aligned} \tag{A.44}$$

Using the same proof idea as in the previous lemma, we show by induction that for $1 \leq k \leq K$, there exists matrices $A_k, B_k \in \mathbb{R}^{d\times d}$ with $\|\!|A_k|\!\|_2 \leq \frac{1}{6}$ and $\|\!|B_k|\!\|_2 \leq \frac{1}{8}$ such that

$$\frac{\partial}{\partial y}F_k(G(x,y),y) = (\mathbb{I}_d - A_k) + k\eta\left(\mathbb{I}_d - B_k\right)\mathbf{J}_y G(x,y). \tag{A.45}$$

Case $k = 1$: The case $k = 1$ can be easily checked according to equation (A.44), we have

$$\frac{\partial}{\partial y} F_1\left(G(x,y), y\right) = \mathbb{I}_d - \frac{\eta^2}{2}\nabla^2 f(y) + \eta \mathbf{J}_y G(x,y)$$

It is sufficient to set $A_1 = \frac{\eta^2}{2}\nabla^2 f(y)$ and $B_1 = 0$.

Case k to $k+1$: Assume the statement is verified until $k \geq 1$. For $k+1 \leq K$, according to equation (A.44), we have

$$\begin{aligned}
&\frac{\partial}{\partial y} F_{k+1}(G(x,y), y) \\
&= \mathbb{I}_d + (k+1)\eta \mathbf{J}_y G(x,y) \\
&- \frac{(k+1)\eta^2}{2}\nabla^2 f(y) - \eta^2 \sum_{j=1}^{k} (k+1-j)\nabla^2 f(F_j(G(x,y),y)) \frac{\partial}{\partial y} F_j(G(x,y),y) \\
&= \mathbb{I}_d - \frac{(k+1)\eta^2}{2}\nabla^2 f(y) + (k+1)\eta \mathbf{J}_y G(x,y) \\
&\quad - \eta^2 \sum_{j=1}^{k} (k+1-j)\nabla^2 f(F_j(G(x,y),y)) \left((\mathbb{I}_d - A_j) + j\eta \left(\mathbb{I}_d - B_j\right) \mathbf{J}_y G(x,y)\right) \\
&= \mathbb{I}_d - \frac{(k+1)\eta^2}{2}\nabla^2 f(y) - \eta^2 \sum_{j=1}^{k} (k+1-j)\nabla^2 f(F_j(G(x,y),y))(\mathbb{I}_d - A_j) \\
&\quad + (k+1)\eta \mathbf{J}_y G(x,y) - \eta^2 \sum_{j=1}^{k} (k+1-j)\nabla^2 f(F_j(G(x,y),y)) \left(j\eta \left(\mathbb{I}_d - B_j\right) \mathbf{J}_y G(x,y)\right)
\end{aligned}$$

To conclude, it suffices to note the following values of A_{k+1} and B_{k+1}:

$$A_{k+1} = \frac{(k+1)\eta^2}{2}\nabla^2 f(y) + \eta^2 \sum_{j=1}^{k} (k+1-j)\nabla^2 f(F_j(G(x,y),y))(\mathbb{I}_d - A_j), \quad \text{and}$$

$$B_{k+1} = \frac{1}{k+1}\eta^2 \sum_{j=1}^{k} (k+1-j) j \nabla^2 f(F_j(G(x,y),y)) \left(\mathbb{I}_d - B_j\right).$$

We now have the following operator norm bounds:

$$\|\!| A_{k+1} |\!\|_2 \leq \frac{k+1}{2}\eta^2 L + \eta^2 \sum_{j=1}^{k} (k+1-j) L (1 + \frac{1}{6}) \leq \frac{7}{12}(k+1)^2 \eta^2 L \leq \frac{1}{6}, \quad \text{and}$$

$$\|\!| B_{k+1} |\!\|_2 \leq \frac{1}{k+1}\eta^2 (1 + \frac{1}{8}) L \sum_{j=1}^{k} (k+1-j) j = \frac{9}{8 \cdot 6} k(k-1)\eta^2 L \leq \frac{1}{8}.$$

This concludes the proof of equation (A.45). As a particular case, for $k = K$, we observe that

$$F_K\left(G(x,y),y\right) = x.$$

Plugging it into equation (A.45), we obtain that

$$\mathbf{J}_y G(x,y) = \frac{1}{K\eta}\left(\mathbb{I}_d - B_K\right)^{-1}\left(\mathbb{I}_d - A_K\right) \quad \Longrightarrow \quad \left\|\mathbf{J}_y G(x,y)\right\|_2 \leq \frac{4}{3K\eta}.$$

Plugging the bound on $\left\|\mathbf{J}_y G(x,y)\right\|_2$ back to equation (A.45) for other k, we obtain

$$\left\|\frac{\partial}{\partial y} F_k(G(x,y),y)\right\|_2 \leq 3.$$

This concludes the proof of Lemma 8.

Proof of Lemma 9

Recall that the backward mapping G is defined implicitly as

$$x = y + K\eta G(x,y) - \frac{K\eta^2}{2}\nabla f(y) - \eta^2 \sum_{k=1}^{K-1}(K-k)\nabla f\left(F_k(G(x,y),y)\right). \tag{A.46}$$

First we check the derivatives of $F_k(G(x,y),y)$. Since $F_k(G(x,y),y)$ satisfies

$$F_k(G(x,y),y) = y + k\eta G(x,y) - \frac{k\eta^2}{2}\nabla f(y) - \eta^2 \sum_{j=1}^{k-1}(k-j)\nabla f(F_j(G(x,y),y)),$$

we have

$$\frac{\partial}{\partial x} F_k(G(x,y),y) = k\eta \mathbf{J}_x G(x,y) - \eta^2 \sum_{j=1}^{k-1}(k-j)\nabla^2 f(F_j(G(x,y),y))\frac{\partial}{\partial x} F_j(G(x,y),y). \tag{A.47}$$

Similar to the proof of equation (A.42), we show by induction (proof omitted) that for $1 \leq k \leq K$, there exists matrices $\tilde{Q}_k \in \mathbb{R}^{d\times d}$ with $\left\|\tilde{Q}_k\right\|_2 \leq \frac{1}{2}$ such that

$$\frac{\partial}{\partial x} F_k(G(x,y),y) = k\eta\left(\mathbb{I}_d - \tilde{Q}_k\right)\mathbf{J}_x G(x,y). \tag{A.48}$$

Then, by taking another derivative with respect to y_i in equation (A.47), we obtain

$$\begin{aligned}
&\frac{\partial\partial}{\partial x \partial y_i} F_k(G(x,y),y) \\
&= k\eta \mathbf{J}_{xy_i} G(x,y) \\
&- \eta^2 \sum_{j=1}^{k-1} (k-j) \Bigg\{ \quad \nabla^3 f_{F_j(G(x,y),y)} \left(\frac{\partial F_j(G(x,y),y)}{\partial y_i}, \mathbb{I}_d, \mathbb{I}_d \right) \frac{\partial}{\partial x} F_j(G(x,y),y) \\
&\qquad\qquad + \nabla^2 f_{F_j(G(x,y),y)} \frac{\partial\partial}{\partial x \partial y_i} F_j(G(x,y),y) \Bigg\}
\end{aligned} \tag{A.49}$$

Now we show by induction that for $1 \le k \le K$, for any $\alpha \in \mathbb{R}^d$, we have

$$\begin{aligned}
\left\| \sum_{i=1}^d \alpha_i \left(\frac{\partial\partial}{\partial x \partial y_i} F_k(G(x,y),y) \mathbf{J}_x G(x,y)^{-1} \right) \right\|_2 &\le 2k\eta \left\| \sum_{i=1}^d \alpha_i \left(\mathbf{J}_{xy_i} G(x,y) \mathbf{J}_x G(x,y)^{-1} \right) \right\|_2 \\
&\quad + 2 \|\alpha\|_2 k^3 \eta^3 L_{\mathrm{H}}.
\end{aligned} \tag{A.50}$$

Case $k = 1$: We first examine the case $k = 1$. According to equation (A.49), we have

$$\sum_{i=1}^d \alpha_i \left(\frac{\partial\partial}{\partial x \partial y_i} F_1(G(x,y),y) \mathbf{J}_x G(x,y)^{-1} \right) = \eta \sum_{i=1}^d \alpha_i \left(\mathbf{J}_{xy_i} G(x,y) \mathbf{J}_x G(x,y)^{-1} \right).$$

The statement in equation (A.50) is easily verified for $k = 1$.

Case k to $k+1$: Assume the statement (A.50) is verified until k. For $k+1 \le K$, according to equation (A.49), we have

$$\begin{aligned}
&\sum_{i=1}^d \alpha_i \left(\frac{\partial\partial}{\partial x \partial y_i} F_{k+1}(G(x,y),y) \mathbf{J}_x G(x,y)^{-1} \right) \\
&= (k+1)\eta \sum_{i=1}^d \alpha_i \left(\mathbf{J}_{xy_i} G(x,y) \mathbf{J}_x G(x,y)^{-1} \right) \\
&\quad - \eta^2 \sum_{j=1}^k (k+1-j) \left\{ \nabla^3 f_{F_j(G(x,y),y)} \left(\sum_{i=1}^d \alpha_i \frac{\partial F_j(G(x,y),y)}{\partial y_i}, \mathbb{I}_d, \mathbb{I}_d \right) \right. \\
&\left. \cdot \frac{\partial}{\partial x} F_j(G(x,y),y) \mathbf{J}_x G(x,y)^{-1} \right\} \\
&\quad - \eta^2 \sum_{j=1}^k (k+1-j) \nabla^2 f_{F_j(G(x,y),y)} \sum_{i=1}^d \alpha_i \left(\frac{\partial\partial}{\partial x \partial y_i} F_j(G(x,y),y) \mathbf{J}_x G(x,y)^{-1} \right).
\end{aligned}$$

In the last equality, we have used the fact that $\nabla^3 f_{F_j(G(x,y),y)}$ is a multilinear form to enter the coefficients α_i in the tensor. Let

$$M_\alpha = \left\| \sum_{i=1}^{d} \alpha_i \left(\mathbf{J}_{xy_i} G(x,y) \mathbf{J}_x G(x,y)^{-1} \right) \right\|_2 .$$

Applying the hypothesis of the induction, we obtain

$$\begin{aligned}
&\left\| \sum_{i=1}^{d} \alpha_i \left(\frac{\partial\partial}{\partial x \partial y_i} F_{k+1}(G(x,y),y) \mathbf{J}_x G(x,y)^{-1} \right) \right\|_2 \\
&\overset{(i)}{\leq} (k+1)\eta M_\alpha + \eta^2 \sum_{j=1}^{k} 4(k+1-j) j L_{\mathrm{H}} \|\alpha\|_2 + \eta^2 \sum_{j=1}^{k} (k+1-j) L \left(2j\eta M + 2\|\alpha\|_2 j^3 \eta^3 L_{\mathrm{H}} \right) \\
&\leq 2(k+1)\eta M_\alpha + 2\|\alpha\|_2 (k+1)^3 \eta^3 L_{\mathrm{H}}.
\end{aligned}$$

The first inequality (i) used the second part of Lemma 8 to bound $\left\| \frac{\partial}{\partial} F_k(G(x,y),y) \right\|_2$. This completes the induction. As a particular case for $k = K$, we note that

$$F_K(G(x,y),y) = F(G(x,y),y) = x,$$

and equation (A.49) for $k = K$ gives

$$\begin{aligned}
0 = {} & K\eta \mathbf{J}_{xy_i} G(x,y) \\
& - \eta^2 \sum_{j=1}^{K-1} (K-j) \Bigg\{ \nabla^3 f_{F_j(G(x,y),y)} \left(\frac{\partial F_j(G(x,y),y)}{\partial y_i}, \mathbb{I}_d, \mathbb{I}_d \right) \frac{\partial}{\partial x} F_j(G(x,y),y) \\
& + \nabla^2 f_{F_j(G(x,y),y)} \frac{\partial\partial}{\partial x \partial y_i} F_j(G(x,y),y) \Bigg\}.
\end{aligned}$$

Using the bound in equation (A.50), we have

$$\begin{aligned}
& K\eta \left\| \sum_{i=1}^{d} \alpha_i \mathbf{J}_{xy_i} G(x,y) \mathbf{J}_x G(x,y)^{-1} \right\|_2 \\
& \leq \|\alpha\|_2 K^3 \eta^3 L_{\mathrm{H}} + \frac{1}{2} K\eta \left\| \sum_{i=1}^{d} \alpha_i \mathbf{J}_{xy_i} G(x,y) \mathbf{J}_x G(x,y)^{-1} \right\|_2 .
\end{aligned}$$

Hence, we obtain

$$\operatorname{trace} \left(\sum_{i=1}^{d} \alpha_i \mathbf{J}_{xy_i} G(x,y) \mathbf{J}_x G(x,y)^{-1} \right) \leq 2d \|\alpha\|_2 K^2 \eta^2 L_{\mathrm{H}}.$$

This is valid for any $\alpha \in \mathbb{R}^d$, as a consequence, we have

$$\left\| \begin{bmatrix} \operatorname{trace} \left([\mathbf{J}_x G(x,q_0)]^{-1} \mathbf{J}_{xy_1} G(x,q_0) \right) \\ \vdots \\ \operatorname{trace} \left([\mathbf{J}_x G(x,q_0)]^{-1} \mathbf{J}_{xy_d} G(x,q_0) \right) \end{bmatrix} \right\|_2 \leq 2dK^2 \eta^2 L_{\mathrm{H}}.$$

This concludes the proof of Lemma 9.

Proof of Lemma 10

We first show equation (3.30b) by induction. Then equation (3.30a) is a direct consequence of equation (3.30b) by summing k terms together.

Case $k = 0$: We first examine the case $k = 0$. According to the definition of F_k in equation (3.26b), we have

$$F_1(p_0, q_0) = q_0 + \eta p_0 - \frac{\eta^2}{2}\nabla f(q_0).$$

Then the case $k = 0$ is verified automatically via triangle inequality,

$$\|F_1(p_0, q_0) - q_0\|_2 \leq \eta \|p_0\|_2 + \frac{\eta^2}{2} \|\nabla f(q_0)\|_2 .$$

Case k to $k+1$: Assume that the statement is verified until $k \geq 0$. For $k+1$, using F_j as the shorthand for $F_j(p_0, q_0)$, we obtain

$$\begin{aligned} &F_{k+2} - F_{k+1} \\ =&\eta p_0 - \frac{\eta^2}{2}\nabla f(q_0) - \eta^2 \sum_{j=1}^{k+1} \nabla f(F_j). \end{aligned}$$

Taking the norm, we have

$$\begin{aligned} \|F_{k+2} - F_{k+1}\|_2 &\leq \eta \|p_0\|_2 + \frac{(2k+3)\eta^2}{2} \|\nabla f(q_0)\|_2 + \eta^2 \sum_{j=1}^{k+1} \|\nabla f(F_j) - \nabla f(q_0)\|_2 \\ &\overset{(i)}{\leq} \eta \|p_0\|_2 + \frac{(2k+3)\eta^2}{2} \|\nabla f(q_0)\|_2 + \eta^2 \sum_{j=1}^{k+1} \sum_{l=0}^{j} \|\nabla f(F_{l+1}) - \nabla f(F_l)\|_2 \\ &\overset{(ii)}{\leq} \eta \|p_0\|_2 + \frac{(2k+3)\eta^2}{2} \|\nabla f(q_0)\|_2 + \eta^2 L \sum_{j=1}^{k+1} \sum_{l=0}^{j} \|F_{l+1} - F_l\|_2 \\ &\overset{(iii)}{\leq} \eta \|p_0\|_2 + \frac{(2k+3)\eta^2}{2} \|\nabla f(q_0)\|_2 \\ &+ \eta^2 L \sum_{j=1}^{k+1} \sum_{l=0}^{j} \left(2\eta \|p\|_2 + 2(l+1)\eta^2 \|\nabla f(q_0)\|_2\right) \\ &\overset{(iv)}{\leq} 2\eta \|p_0\|_2 + (2k+2)\eta^2 \|\nabla f(q_0)\|_2 . \end{aligned}$$

Inequality (i) uses triangular inequality. Inequality (ii) uses L-smoothness. Inequality (iii) applies the hypothesis of the induction and inequalities relies on the condition $K^2\eta^2 \leq \frac{1}{4L}$. This completes the induction.

A.2 Proof of Corollary 3

In order to prove Corollary 3, we first state a more general corollary of Theorem 3 that does not specify the explicit choice of step size η and leapfrog steps K. Then we specify two choices of the initial distribution μ_0 and hyper-parameters (K, η) to obtain part (a) and part (b) of Corollary 3.

Corollary 6. *Consider an (L, L_H, m)-strongly log-concave target distribution Π^* (cf. Assumption (B)). Fix $s = \frac{\epsilon^2}{2\varpi}$. Then the $\frac{1}{2}$-lazy HMC algorithm with initial distribution $\mu_\dagger = \mathcal{N}(x^*, \frac{1}{L}\mathbb{I}_d)$, step size η and leapfrog steps K chosen under the condition*

$$\eta^2 \leq \frac{1}{cL} \min \left\{ \frac{1}{K^2 d^{\frac{1}{2}}}, \frac{1}{K^2 d^{\frac{2}{3}}} \frac{L}{L_H^{\frac{2}{3}}}, \frac{1}{K d^{\frac{1}{2}}}, \frac{1}{K^{\frac{2}{3}} d^{\frac{2}{3}} \kappa^{\frac{1}{3}} r(s)^{\frac{2}{3}}}, \frac{1}{K d^{\frac{1}{2}} \kappa^{\frac{1}{2}} r(s)}, \right.$$
$$\left. \frac{1}{K^{\frac{2}{3}} d} \frac{L}{L_H^{\frac{2}{3}}}, \frac{1}{K^{\frac{4}{3}} d^{\frac{1}{2}} \kappa^{\frac{1}{2}} r(s)} \left(\frac{L}{L_H^{\frac{2}{3}}} \right)^{\frac{1}{2}} \right\} \tag{A.51}$$

satisfies the mixing time bounds

$$\tau_2^{HMC}(\epsilon; \mu_0) \leq c \cdot \max \left\{ \log \varpi, \frac{1}{K^2 \eta^2 m} \log \left(\frac{d \log \kappa}{\epsilon} \right) \right\}.$$

Proof of part (a) in Corollary 3: Taking the hyper-parameters $K = d^{\frac{1}{4}}$ and $\eta = \eta_{\text{warm}}$ in equation (3.7b), we verify that η satisfies the condition (A.51). Given the warmness parameter $\varpi = O\left(\exp\left(d^{\frac{2}{3}}\kappa\right)\right)$, we have

$$\frac{1}{K^2 \eta^2 m} \geq \log(\varpi).$$

Plugging in the choice of K and η into Corollary 6, we obtain the desired result.

Proof of part (b) in Corollary 3: We notice that the initial distribution $\mu_\dagger = \mathcal{N}(x^\star, \frac{1}{L}\mathbb{I}_d)$ is $\kappa^{d/2}$-warm (see Corollary 1 in [58]). It is sufficient to plug in the hyper-parameters $K = \kappa^{\frac{3}{4}}$ and $\eta = \eta_{\text{feasible}}$ into Corollary 6 to obtain the desired result.

Now we turn back to prove Corollary 6. In order to prove Corollary 6, we require the the following lemma, which relates a (L, L_{H}, m)-strongly-logconcave target distribution to a regular target distribution.

Lemma 20. *An (L, L_H, m)-strongly log-concave distribution is $(L, L_H, s, \psi_{\frac{1}{2}}, M)$-general with high mass set $\Omega = \mathcal{R}_s$, log-isoperimetric constant $\psi_{\frac{1}{2}} = m^{-\frac{1}{2}}$ and $M = L\left(\frac{d}{m}\right)^{\frac{1}{2}} r(s)$, where the radius is defined in equation (3.7a) and the convex measurable set $\mathcal{R}_s$ defined in equation (3.25).*

Taking Lemma 20 as given, Corollary 6 is a direct consequence of Theorem 3 by plugging the specific values of $(\Omega, \psi_{\frac{1}{2}}, M)$ as a function of strong convexity parameter m. The optimal choices of step-size η and leapfrog steps K in Corollary 6 are discussed in Appendix A.4.1.

We now proceed to prove Lemma 20.

A.2.1 Proof of Lemma 20

We now prove Lemma 20, which shows that any (L, L_{H}, m)-strongly-logconcave target distribution is in fact $(L, L_{\mathrm{H}}, s, \psi_{\frac{1}{2}}, M)$-regular.

First, we set Ω to $\mathcal{R}_s$ as defined in equation (3.25). It is known that this ball has probability under the target distribution lower bounded as $\Pi^*(\mathcal{R}_s) \geq 1 - s$ (e.g. Lemma 1 in the paper [58]). Second, the gradient bound is a consequence of the bounded domain. For any $x \in \mathcal{R}_s$, we have

$$\|\nabla f(x)\|_2 = \|\nabla f(x) - \nabla f(x^\star)\|_2 \leq L \|x - x^\star\|_2 \leq L \left(\frac{d}{m}\right)^{\frac{1}{2}} r(s). \tag{A.52}$$

Third, we make use of a logarithmic isoperimetric inequality for log-concave distribution. We note that the logarithmic isoperimetric inequality has been introduced in Kannan et al. [94] for the uniform distribution on convex body and in Lee and Vempala [111] for log-concave distribution with a diameter. We extend this inequality to strongly log-concave distribution on $\mathbb{R}^d$ following a similar road-map and provide explicit constants.

Improved logarithmic isoperimetric inequality We now state the improved logarithmic isoperimetric inequality for strongly log-concave distributions.

Lemma 21. *Let γ denote the density of the standard Gaussian distribution $\mathcal{N}(0, \sigma^2 \mathbb{I}_d)$, and let Π^* be a distribution with density $\pi^* = q \cdot \gamma$, where q is a log-concave function. Then for any partition S_1, S_2, S_3 of $\mathbb{R}^d$, we have*

$$\Pi^*(S_3) \geq \frac{d(S_1, S_2)}{2\sigma} \min\{\Pi^*(S_1), \Pi^*(S_2)\} \log^{\frac{1}{2}} \left(1 + \frac{1}{\min\{\Pi^*(S_1), \Pi^*(S_2)\}}\right). \tag{A.53}$$

See Appendix A.2.2 for the proof.

Taking Lemma 21 as given for the moment, we turn to prove the logarithmic isoperimetric inequality for the Ω-restricted distribution Π^*_Ω with density

$$\pi^*_\Omega(x) = \frac{\pi^*(x)\mathbf{1}_\Omega(x)}{\Pi^*(\Omega)}.$$

Since f is m-strongly convex, the function $x \to f(x) - \frac{m}{2}\|x - x^\star\|_2^2$ is convex. Noting that the class of log-concave function is closed under multiplication and that the

indicator function $\mathbf{1}_\Omega$ is log-concave, we conclude that the restricted density π^*_Ω can be expressed as a product of a log-concave density and the density of the Gaussian distribution $\mathcal{N}(x^\star, \frac{1}{m}\mathbb{I}_d)$. Applying Lemma 21 with $\sigma = \left(\frac{1}{m}\right)^{\frac{1}{2}}$, we obtain the desired logarithmic isoperimetric inequality with $\psi_{\frac{1}{2}} = \left(\frac{1}{m}\right)^{\frac{1}{2}}$, which concludes the proof of Lemma 20.

A.2.2 Proof of Lemma 21

The main tool for proving general isoperimetric inequalities is the localization lemma introduced by Lovász and Simonovits [118]. Similar result for the infinitesimal version of equation (A.53) have appeared as Theorem 1.1 in the paper [106] and Theorem 30 in the paper [111]. Intuitively, the localization lemma reduces a high-dimensional isoperimetric inequality to a one-dimensional inequality which is much easier to verify directly. In a few key steps, the proof follows a similar road map as the proof of logarithmic Cheeger inequality [94].

We first state an additional lemma that comes in handy for the proof.

Lemma 22. *Let γ be the density of the one-dimensional Gaussian distribution $\mathcal{N}(\nu, \sigma^2)$ with mean ν and variance σ^2. Let ρ be a one-dimensional distribution with density given by $\rho = q \cdot \gamma$, where q is a log-concave function supported on $[0,1]$. Let J_1, J_2, J_3 partition $[0,1]$, then*

$$\rho(J_3) \geq \frac{d(J_1, J_2)}{2\sigma} \min\{\rho(J_1), \rho(J_2)\} \log^{\frac{1}{2}}\left(1 + \frac{1}{\min\{\rho(J_1), \rho(J_2)\}}\right). \tag{A.54}$$

See Appendix A.2.3 for the proof.

We now turn to proving Lemma 21 via contradiction: We assume that the claim (A.53) is not true for some partition, and then using well known localization techniques, we construct a one-dimensional distribution that violates Lemma 22 resulting in a contradiction.

Suppose that there exists a partition S_1, S_2, S_3 of $\mathbb{R}^d$, such that

$$\Pi^*(S_3) < \frac{d(S_1, S_2)}{2\sigma} \min\{\Pi^*(S_1), \Pi^*(S_2)\} \log^{\frac{1}{2}}\left(1 + \frac{1}{\min\{\Pi^*(S_1), \Pi^*(S_2)\}}\right). \tag{A.55}$$

Let $\nu > 0$ denote a sufficiently small number (to be specified exactly later), such that $\nu < \min\{\Pi^*(S_1), \Pi^*(S_2)\}$.

We now explain the construction of the one-dimensional density that is crucial for the rest of the argument. We define two functions $g : \mathcal{X} \to \mathbb{R}$ and $h : \mathcal{X} \to \mathbb{R}$ as follows

$$g(x) = \frac{\pi^*(x) \cdot \mathbf{1}_{S_1}(x)}{\Pi^*(S_1) - \nu} - \pi^*(x) \quad \text{and} \quad h(x) = \frac{\pi^*(x) \cdot \mathbf{1}_{S_2}(x)}{\Pi^*(S_2) - \nu} - \pi^*(x).$$

Clearly, we have

$$\int_{\mathcal{X}} g(x)dx > 0 \quad \text{and} \quad \int_{\mathcal{X}} h(x)dx > 0.$$

By the localization lemma (Lemma 2.5 in the paper [118]; see the corrected form stated as Lemma 2.1 in the paper [95]), there exist two points $a \in \mathbb{R}^d, b \in \mathbb{R}^d$ and a linear function $l : [0,1] \to \mathbb{R}_+$, such that

$$\int_0^1 l(t)^{d-1} g\left((1-t)a + tb\right) dt > 0 \quad \text{and} \quad \int_0^1 l(t)^{d-1} h\left((1-t)a + tb\right) dt > 0. \tag{A.56}$$

Define the one-dimensional density $\rho : [0,1] \to \mathbb{R}^+$ and the sets $J_i, i \in \{1,2,3\}$ as follows:

$$\rho(t) = \frac{l(t)^{d-1}\pi^*\left((1-t)a + tb\right)}{\int_0^1 l(u)^{d-1}\pi^*\left((1-u)a + ub\right) du}, \quad \text{and} \tag{A.57}$$

$$J_i = \{t \in [0,1] \mid (1-t)a + tb \in S_i\} \quad \text{for } i \in \{1,2,3\}. \tag{A.58}$$

We now show how the hypothesis (A.55) leads to a contradiction for the density ρ. Plugging in the definiton of g and h into equation (A.56), we find that

$$\rho(J_1) > \Pi^*(S_1) - \nu \quad \text{and} \quad \rho(J_2) > \Pi^*(S_2) - \nu.$$

Since J_1, J_2, J_3 partition $[0,1]$, it follows that

$$\rho(J_3) < \Pi^*(S_3) + 2\nu.$$

Since the function $x \mapsto x \log^{\frac{1}{2}}(1 + 1/x)$ is monotonically increasing on $[0,1]$, we have

$$\begin{aligned}
&\frac{d(S_1,S_2)}{2\sigma} \min\left\{\rho(J_1), \rho(J_2)\right\} \log^{\frac{1}{2}}\left(1 + \frac{1}{\min\left\{\rho(J_1), \rho(J_2)\right\}}\right) - \rho(J_3) \\
&\quad \geq \frac{d(S_1,S_2)}{2\sigma} \min\left\{\left(\rho(S_1) - \nu\right), \left(\rho(S_2) - \nu\right)\right\} \cdot \\
&\qquad\qquad \log^{\frac{1}{2}}\left(1 + \frac{1}{\min\left\{\left(\rho(S_1) - \nu\right), \left(\rho(S_2) - \nu\right)\right\}}\right) - \left(\rho(S_3) + 2\nu\right)
\end{aligned}$$

The hypothesis (A.55) of the contradiction implies that we can find ν sufficiently small such that the RHS in the inequality above will be strictly positive. Consequently, we obtain

$$\frac{d(S_1,S_2)}{2\sigma} \min\left\{\rho(J_1), \rho(J_2)\right\} \log^{\frac{1}{2}}\left(1 + \frac{1}{\min\left\{\rho(J_1), \rho(J_2)\right\}}\right) > \rho(J_3). \tag{A.59}$$

Additionally, for $t_1 \in J_1, t_2 \in J_2$, we have $(1-t_1)a + t_1 b \in S_1$ and $(1-t_2)a + t_2 b \in S_2$. As a result, we have

$$|t_1 - t_2| = \frac{1}{\|b-a\|_2} \left\|\left[(1-t_1)a + t_1 b\right] - \left[(1-t_2)a + t_2 b\right]\right\|_2 \geq \frac{1}{\|b-a\|_2} d(S_1, S_2),$$

which implies that

$$d(J_1, J_2) \geq \frac{1}{\|b-a\|_2} d(S_1, S_2). \tag{A.60}$$

Combining equations (A.59) and (A.60), we obtain that

$$\frac{\|b-a\|_2 \cdot d(J_1, J_2)}{2\sigma} \min\{\rho(J_1), \rho(J_2)\} \log^{\frac{1}{2}} \left(1 + \frac{1}{\min\{\rho(J_1), \rho(J_2)\}}\right) > \rho(J_3), \tag{A.61}$$

which contradicts Lemma 22. Indeed, this contradiction is immediate once we note that the new density ρ can also be written as a product of log-concave density and a Gaussian density with variance $\frac{\sigma^2}{\|b-a\|_2^2}$.

A.2.3 Proof of Lemma 22

We split the proof into three cases. Each one is more general than the previous one. First, we consider the case when q is a constant function on $[0,1]$ and the sets J_1, J_2, J_3 are all intervals. In the second case, we consider a general log-concave q supported on $[0,1]$ while we still assume that the sets J_1, J_2, J_3 are all intervals. Finally, in the most general case, we consider a general log-concave q supported on $[0,1]$ and J_1, J_2, J_3 consist of an arbitrary partition of $[0,1]$. The proof idea follows roughly that of Theorem 4.6 in Kannan et al. [94].

Our proof makes use of the Gaussian isoperimetric inequality which we now state (see e.g., equation (1.2) in [16]): Let Γ denote the standard univariate Gaussian distribution and let ϕ_Γ and Φ_Γ^{-1} denote its density and inverse cumulative distribution function respectively. Given a measurable set $A \subset \mathbb{R}$, define its Γ-perimeter $\Gamma^+(A)$ as

$$\Gamma^+(A) = \liminf_{h \to 0^+} \frac{\Gamma(A+h) - \Gamma(A)}{h},$$

where $A + h = \{t \in \mathbb{R} \mid \exists a \in A, |t-a| < h\}$ denotes an h-neighborhood of A. Then, we have

$$\Gamma^+(A) \geq \phi_\Gamma(\Phi_\Gamma^{-1}(\Gamma(A))), \tag{A.62}$$

Furthermore, standard Gaussian tail bounds[2] estimate imply that

$$\phi_\Gamma(\Phi_\Gamma^{-1}(t)) \geq \frac{1}{2} t \log^{\frac{1}{2}} \left(1 + \frac{1}{t}\right), \quad \text{for } t \in (0, \frac{1}{2}]. \tag{A.63}$$

[2]E.g., see the discussion before equation 1 in the paper [9]. The constant 1/2 was estimated by plotting the continuous function on the left hand side via Mathematica.

Case 1: First, we consider the case when the function q is constant on $[0,1]$ and all of the sets J_1, J_2, J_3 are intervals. Without loss of generality, we can shift and scale the density function by changing the domain, and assume that the density ρ is of the form $\rho(t) \propto e^{-\frac{t^2}{2}} \mathbf{1}_{[a,d]}$. Additionally, we can assume that J_1, J_2, J_3 are of the form

$$J_1 = [a, b), \quad J_3 = [b, c], \quad \text{and} \quad J_2 = (c, d], \tag{A.64}$$

because the case when J_3 is not in the middle is a trivial case.

Applying the inequalities (A.62) and (A.63) with $A = J_2 = (c, d]$, we obtain that

$$\phi_\gamma(c) = \Gamma^+(J_2) \geq \phi_\gamma(\Phi_\gamma^{-1}(\Gamma(J_2))) \geq \frac{\Gamma(J_2)}{2} \log^{\frac{1}{2}} \left(1 + \frac{1}{\Gamma(J_2)}\right). \tag{A.65}$$

Note that $\rho(t) = \frac{\phi_\gamma(t)}{\Phi_\gamma(d) - \Phi_\gamma(a)} \mathbf{1}_{[a,d]}(t)$ and $\rho(J_2) = \frac{\Gamma(J_2)}{\Phi_\gamma(d) - \Phi_\gamma(a)}$. We have

$$\begin{aligned}
\rho(J_3) &= \int_b^c \rho(t) dt \\
&\geq (c - b) \cdot \rho(c) \\
&= (c - b) \frac{\phi_\gamma(c)}{\Phi_\gamma(d) - \Phi_\gamma(a)} \\
&\overset{(i)}{\geq} \frac{(c - b)}{2} \frac{\Gamma(J_2)}{\Phi_\gamma(d) - \Phi_\gamma(a)} \log^{\frac{1}{2}} \left(1 + \frac{1}{\Gamma(J_2)}\right) \\
&\overset{(ii)}{\geq} \frac{c - b}{2} \rho(J_2) \log^{\frac{1}{2}} \left(1 + \frac{\Phi_\gamma(d) - \Phi_\gamma(a)}{\Gamma(J_2)}\right) \\
&\overset{(iii)}{=} \frac{c - b}{2} \rho(J_2) \log^{\frac{1}{2}} \left(1 + \frac{1}{\rho(J_2)}\right) \\
&\overset{(iv)}{\geq} \frac{c - b}{2} \min\{\rho(J_1), \rho(J_2)\} \log^{\frac{1}{2}} \left(1 + \frac{1}{\min\{\rho(J_1), \rho(J_2)\}}\right),
\end{aligned}$$

where step (i) follows from the bound (A.65) and step (ii) follows from the relationship between ρ and Γ and the facts that log is an increasing function and that $\Phi_\gamma(d) - \Phi_\gamma(a) \leq 1$. Step (iii) follows from the definition of ρ and finally step (iv) follows from the increasing nature of the map $t \mapsto t \log^{1/2}\left(1 + \frac{1}{t}\right)$. This concludes the argument for Case 1.

Case 2: We now consider the case when q is a general log-concave function on $[0,1]$ and J_1, J_2, J_3 are all intervals. Again we can assume that J_1, J_2, J_3 are of the form (A.64), i.e., they are given by $J_1 = [a, b)$, $J_3 = [b, c]$, and $J_2 = (c, d]$.

We consider an exponential function $h(t) = \alpha e^{\beta t - \frac{t^2}{2\sigma^2}}$ such that $h(b) = q(b)$ and $h(c) = q(c)$.[3] Define $Q(t_1, t_2) = \int_{t_1}^{t_2} q(t) dt$ and $H(t_1, t_2) = \int_{t_1}^{t_2} h(t) dt$. Then since q has

[3]This idea of introducing exponential function appeared in Corollary 6.2 of Kannan et al. [94].

an extra log-concave component compared to h, we have

$$H(a,b) \geq Q(a,b), \quad H(c,d) \geq Q(c,d), \text{ but } H(b,c) \leq Q(b,c). \tag{A.66}$$

Using the individual bounds in equation (A.66), we have

$$\frac{1}{H(a,b)} + \frac{1}{H(c,d)} + \frac{H(b,c)}{H(a,b)H(c,d)} \leq \frac{1}{Q(a,b)} + \frac{1}{Q(c,d)} + \frac{Q(b,c)}{Q(a,b)Q(c,d)}.$$

Consequently, we obtain

$$\frac{H(a,b)H(c,d)}{H(a,d)} \geq \frac{Q(a,b)Q(c,d)}{Q(a,d)}. \tag{A.67}$$

Using the individual bounds in equation (A.66) again, we have

$$\frac{H(a,b)}{H(b,c)} + \frac{H(c,d)}{H(b,c)} \geq \frac{Q(a,b)}{Q(b,c)} + \frac{Q(c,d)}{Q(b,c)},$$

Consequently, we obtain

$$\frac{H(b,c)}{H(a,d)} \leq \frac{Q(b,c)}{Q(a,d)}. \tag{A.68}$$

Combining equation (A.67) and (A.68), applying that the function $t \mapsto t \log^{\frac{1}{2}}\left(1+\frac{1}{t}\right)$ is increasing, we verify that the inequality (A.54) on ρ can be reduced to the inequality on h. h is Gaussian when restricted to the interval $[a,d]$, so applying the result in the case 1 we conclude the case 2.

Case 3: Finally, we deal with the general case where J_1, J_2, J_3 each can be union of intervals and q is a general log-concave function on $[0,1]$. We show that this case can be reduced to the case of three intervals, namely, the previous case.

Let $\{(b_i,c_i)\}_{i\in\mathcal{I}}$ be all non-empty maximal intervals contained in J_3. Here the intervals can be either closed, open or half. That is, $(\cdot,\cdot)$ can be $[\cdot,\cdot]$, $]\cdot,\cdot[$, $[\cdot,\cdot[$ or $]\cdot,\cdot]$. For an interval (b_i,c_i), we define its left surround $LS((b_i,c_i))$ as

$$LS((b_i,c_i)) = \begin{cases} 2, \text{ if } \exists x_2 \in J_2, (x_2 \leq b_i) \text{ and } (\nexists x_1 \in J_1, x_2 < x_1 \leq b_i) \\ 1, \text{ if } \exists x_1 \in J_1, (x_1 \leq b_i) \text{ and } (\nexists x_2 \in J_2, x_1 < x_2 \leq b_i) \\ 0, \text{ otherwise .} \end{cases}$$

Similarly, we define $RS((b_i,c_i))$ as

$$RS((b_i,c_i)) = \begin{cases} 2, \text{ if } \exists x_2 \in J_2, (x_2 \geq c_i) \text{ and } (\nexists x_1 \in J_1, x_2 > x_1 \geq c_i) \\ 1, \text{ if } \exists x_1 \in J_1, (x_1 \geq c_i) \text{ and } (\nexists x_2 \in J_2, x_1 > x_2 \geq c_i) \\ 0, \text{ otherwise .} \end{cases}$$

We distinguish two types of intervals. Denote $G_2 \subset \mathcal{I}$ the set containing the indices of all intervals that are surrounded by either 1 or 2 but different.

$$G_2 := \{i \in \mathcal{I} \mid (LS((b_i, c_i)), RS((b_i, c_i))) = (1, 2) \text{ or } (2, 1)\}.$$

Denote $G_1 := \mathcal{I} \setminus G_2$ to be its complement. By the result settled in case 2, for $i \in G_2$, we have

$$\rho([b_i, c_i]) \geq \frac{d(J_1, J_2)}{2\sigma} \rho(I_i) \log^{\frac{1}{2}} \left(1 + \frac{1}{\rho(I_i)}\right)$$

where I_i is either $[a, b_i]$ or $[c_i, d]$. Summing over all $i \in G_2$, we have

$$\begin{aligned} \rho(J_3) \geq \sum_{i \in G_2} \rho([b_i, c_i]) &\geq \frac{d(J_1, J_2)}{2\sigma} \sum_{i \in G_2} \rho(I_i) \log^{\frac{1}{2}} \left(1 + \frac{1}{\rho(I_i)}\right) \\ &\geq \frac{d(J_1, J_2)}{2\sigma} \rho(\cup_{i \in G_2} I_i) \log^{\frac{1}{2}} \left(1 + \frac{1}{\rho(\cup_{i \in G_2} I_i)}\right). \end{aligned} \tag{A.69}$$

The last inequality follows from the sub-additivity of the map: $x \mapsto x \log^{\frac{1}{2}}(1 + x)$, i.e., for $x > 0$ and $y > 0$, we have

$$x \log^{\frac{1}{2}} \left(1 + \frac{1}{x}\right) + y \log^{\frac{1}{2}} \left(1 + \frac{1}{y}\right) \geq (x + y) \log^{\frac{1}{2}} \left(1 + \frac{1}{x + y}\right).$$

Indeed the sub-additivity follows immediately from the following observation:

$$\begin{aligned} &x \log^{\frac{1}{2}} \left(1 + \frac{1}{x}\right) + y \log^{\frac{1}{2}} \left(1 + \frac{1}{y}\right) - (x + y) \log^{\frac{1}{2}} \left(1 + \frac{1}{x + y}\right) \\ &= x \left[\log^{\frac{1}{2}} \left(1 + \frac{1}{x}\right) - \log^{\frac{1}{2}} \left(1 + \frac{1}{x + y}\right)\right] + y \left[\log^{\frac{1}{2}} \left(1 + \frac{1}{y}\right) - \log^{\frac{1}{2}} \left(1 + \frac{1}{x + y}\right)\right] \\ &\geq 0. \end{aligned}$$

Finally, we remark that either J_1 or J_2 is a subset of $\cup_{i \in G_2} I_i$. If not, there exists $u \in J_1 \setminus \cup_{i \in G_2} I_i$ and $v \in J_2 \setminus \cup_{i \in G_2} I_i$, such that u and v are separated by some inverval $(b_{i^*}, c_{i^*}) \subset J_3$ with $i^* \in G_2$. This is contradictory with the fact that either u or v must be included in I_{i^*}. Given equation (A.69), we use the fact that the function $x \mapsto x \log^{\frac{1}{2}} \left(1 + \frac{1}{x}\right)$ is monotonically increasing:

$$\rho(J_3) \geq \frac{d(J_1, J_2)}{2\sigma} \min\{\rho(J_1), \rho(J_2)\} \log^{\frac{1}{2}} \left(1 + \frac{1}{\min\{\rho(J_1), \rho(J_2)\}}\right)$$

to conclude the proof.

A.3 Beyond strongly log-concave

In this appendix, we continue the discussion of mixing time bounds of Metropolized HMC from Section 3.3.2. In the next two subsections, we discuss the case when the target is weakly log-concave distribution or a perturbation of log-concave distribution, respectively.

A.3.1 Weakly log-concave target

The mixing rate in the weakly log-concave case differs depends on further structural assumptions on the density. We now consider two different scenarios where either a bound on fourth moment is known or the covariance of the distribution is well-behaved:

(C) The negative log density of the target distribution is L-smooth (3.6a) and has L_{H}-Lipschitz Hessian (3.6c). Additionally for some point $x^\star$, its fourth moment satisfies the bound

$$\int_{\mathbb{R}^d} \|x - x^\star\|_2^4 \pi^*(x) dx \leq \frac{d^2\nu^2}{L}. \tag{A.70}$$

(D) The negative log density of the target distribution is L-smooth (3.6a) and has L_{H}-Lipschitz Hessian (3.6c). Additionally, its covariance matrix satisfies

$$\left\| \int_{x\in\mathbb{R}^d} (x - \mathbb{E}[x])(x - \mathbb{E}[x])^\top \pi^*(x) dx \right\|_{\mathrm{op}} \leq 1, \tag{A.71}$$

and the norm of the gradient of the negative log density f is bounded by a constant in the ball $\mathbb{B}\left(\mathbb{E}\left[x\right], \log\left(\frac{1}{s}\right) d^{3/4}\right)$ for small enough $s \geq s_0$.

When the distribution satisfies assumption (C) we consider HMC chain with slightly modified target and assume that the μ_0 is ϖ-warm with respect to this modified target distribution (see the discussion after Corollary 7 for details). Moreover, In order to simplify the bounds in the next result, we assume that $L_{\mathrm{H}}^{2/3} = O(L)$. A more general result with without this condition can be derived in a similar fashion.

Corollary 7 (HMC mixing for weakly-log-concave)**.** *Let μ_0 be a ϖ-warm start, $\epsilon \in (0,1)$ be fixed and consider $\frac{1}{2}$-lazy HMC chain with leapfrog steps $K = d^{\frac{1}{2}}$ and step size $\eta^2 = \frac{1}{cLd^{\frac{4}{3}}}$.*

(a) If the distribution satisfies assumption (C), then we have

$$\tau_{\mathrm{TV}}^{HMC}(\epsilon; \mu_0) \leq c \cdot \max\left\{ \log \varpi, \frac{d^{\frac{4}{3}}\nu}{\epsilon} \log\left(\frac{\log \varpi}{\epsilon}\right) \right\}. \tag{A.72}$$

(b) If the distribution satisfies assumption (D) such that $s_0 \leq \frac{\epsilon^2}{2\varpi}$*, then we have*

$$\tau_2^{HMC}(\epsilon; \mu_0) \leq c \cdot d^{\frac{5}{6}} \log\left(\frac{\log \varpi}{\epsilon}\right). \tag{A.73}$$

As an immediate consequence, we obtain that the number of gradient evaluations in the two cases is bounded as

$$\mathcal{B}_1 = \max\left\{d^{\frac{1}{2}} \log \varpi, \frac{d^{\frac{11}{6}}\nu}{\epsilon} \log\left(\frac{\log \varpi}{\epsilon}\right)\right\} \quad \text{and} \quad \mathcal{B}_2 = d^{\frac{4}{3}} \log\left(\frac{\log \varpi}{\epsilon}\right).$$

We remark that the bound $\mathcal{B}_1$ for HMC chain improves upon the bound for number of gradient evaluations required by MALA to mix in a similar set-up. Dwivedi et al. [58] showed that under assumption (C) (without the Lipschitz-Hessian condition), MALA takes $O(\frac{d^2}{\nu\epsilon} \log \frac{\varpi}{\epsilon})$ steps to mix. Since each step of MALA uses one gradient evaluation, our result shows that HMC takes $O(d^{\frac{1}{6}})$ fewer gradient evaluations. On the other hand, when the target satisfies assumption (D), Mangoubi et al. [129] showed that MALA takes $O(d^{\frac{3}{2}} \log \frac{\varpi}{\epsilon})$ steps.[4] Thus even for this case, our result shows that HMC takes $O(d^{\frac{1}{6}})$ fewer gradient evaluations when compared to MALA.

Proof sketch: When the target distribution has a bounded fourth moment (assumption (C)), proceeding as in the paper [43], we can approximate the target distribution Π^* by a strongly log-concave distribution $\tilde{\Pi}$ with density given by

$$\tilde{\pi}(x) = \frac{1}{\int_{\mathbb{R}^d} e^{-\tilde{f}(y)} dy} e^{-\tilde{f}(x)} \quad \text{where } \tilde{f}(x) = f(x) + \frac{\lambda}{2}\|x - x^\star\|_2^2.$$

Setting $\lambda := \frac{2L\epsilon}{d\nu}$ yields that $\tilde{f}$ is $\lambda/2$-strongly convex, $L + \lambda/2$ smooth and L_{H}-Hessian Lipschitz and that the TV distance $d_{\mathrm{TV}}\left(\Pi^*, \tilde{\Pi}\right) \leq \epsilon/2$ is small. The new condition number becomes $\widetilde{\kappa} := 1 + d\nu/\epsilon$. The new logarithmic-isoperimetric constant is $\widetilde{\psi}_{1/2} = (d\nu/(L\epsilon))^{1/2}$. Thus, in order to obtain an ϵ-accurate sample with respect to Π^*, it is sufficient to run HMC chain on the new strongly log-concave distribution $\widetilde{\Pi}$ upto $\epsilon/2$-accuracy. Invoking Corollary 3 for $\widetilde{\Pi}$ and doing some algebra yields the bound (A.72).

For the second case (assumption (D)), Lee et al. [113] showed that when the covariance of Π^* has a bounded operator norm, it satisfies isoperimetry inequality (3.6d) with $\psi_0 \leq O(d^{\frac{1}{4}})$. Moreover, using the Lipschitz concentration [73], we have

$$\mathbb{P}_{x\sim\Pi^*}\left(\|x - \mathbb{E}_{\Pi^*}[x]\|_2 \geq t\psi_0 \cdot \sqrt{d}\right) \leq e^{-ct},$$

[4] Note that the authors of the paper [129] assume an infinity-norm third order smoothness which is a stronger assumption than the L_{H}-Lipschitz Hessian assumption that we made here. Under our setting, the infinity norm third order smoothness is upper bounded by $\sqrt{d}L_{\mathrm{H}}$ and plugging in this bound changes their rate of MALA from $d^{7/6}$ to $d^{3/2}$.

which implies that for $\Omega_s = \mathbb{B}\left(\mathbb{E}_{\Pi^*}[x], \frac{1}{c}\log\left(\frac{1}{s}\right)\psi_0 \cdot \sqrt{d}\right)$, we have $\Pi^*(\Omega_s) \geq 1 - s$. In addition, assuming that the gradient is bounded in this ball Ω_s for $s = \frac{\epsilon^2}{2\varpi}$ enables us to invoke Theorem 3 and obtain the bound (A.73) after plugging in the values of ψ_0, K and η.

A.3.2 Non-log-concave target

We now briefly discuss how our mixing time bounds in Theorem 3 can be applied for distributions whose negative log density may be non-convex. Let Π be a log-concave distribution with negative log density as f and isoperimetric constant ψ_0. Suppose that the target distribution $\widetilde{\Pi}$ is a perturbation of Π with target density $\widetilde{\pi}(x)$ such that $\widetilde{\pi}(x) \propto e^{-f(x)-\xi(x)}$, where the perturbation $\xi : \mathbb{R}^d \to \mathbb{R}$ is uniformly lower bounded by some constant $-b$ with $b \geq 0$. Then it can be shown that the distribution $\widetilde{\Pi}$ satisfies isoperimetric inequality (3.6d) with a constant $\widetilde{\psi}_0 \geq e^{-2b}\psi_0$. For example, such type of a non-log-concave distribution distribution arises when the target distribution is that of a Gaussian mixture model with several components where all the means of different components are close to each other (see e.g. the paper [124]). If a bound on the gradient is also known, Theorem 3 can be applied to obtain a suitable mixing time bound. However deriving explicit bounds in such settings is not the focus of the chapter and thereby we omit the details here.

A.4 Optimal choice for HMC hyper-parameters

In this section, we provide a detailed discussion about the optimal leapfrog steps choice for Metropolized HMC with strongly log-concave target distribution (Corollary 3). We also discuss a few improved convergence rates for Metropolized HMC under additional assumptions on the target distribution. Finally, we compare our results for Metropolized HMC with other versions of HMC namely unadjusted HMC and ODE-solved based HMC in Subsection A.4.2.

A.4.1 Optimal choices for Corollary 6

Corollary 6 provides an implicit condition that the step size η and leapfrog steps K should satisfy and provides a generic mixing time upper bound that depends on the choices made. We claim that the optimal choices of η and K according to Table A.1 lead to the following upper bound on number of gradient evaluations required by HMC to mix to ϵ-tolerance:

$$K \cdot \tau_{\mathrm{TV}}^{\mathrm{HMC}}(\epsilon; \mu_0) \leq O\left(\max\left\{d\kappa^{\frac{3}{4}}, d^{\frac{11}{12}}\kappa, d^{\frac{3}{4}}\kappa^{\frac{5}{4}}, d^{\frac{1}{2}}\kappa^{\frac{3}{2}}\right\} \cdot \log\frac{1}{\epsilon}\right). \tag{A.74}$$

This (upper) bound shows that HMC always requires fewer gradient evaluations when compared to MALA for mixing in total variation distance. However, such a bound

requires a delicate choice of the leap frog steps K and η depending on the condition number κ and the dimension d, which might be difficult to implement in practice. We summarize these optimal choices in Table A.1.

Case	K	η^2
$\kappa \in (0, d^{\frac{1}{3}})$	$\kappa^{\frac{3}{4}}$	$\frac{1}{cL} \cdot d^{-1}\kappa^{-\frac{1}{2}}$
$\kappa \in [d^{\frac{1}{3}}, d^{\frac{2}{3}}]$	$d^{\frac{1}{4}}$	$\frac{1}{cL} \cdot d^{-\frac{7}{6}}$
$\kappa \in (d^{\frac{2}{3}}, d]$	$d^{\frac{3}{4}}\kappa^{-\frac{3}{4}}$	$\frac{1}{cL} \cdot d^{-\frac{3}{2}}\kappa^{\frac{1}{2}}$
$\kappa \in (d, \infty)$	1	$\frac{1}{cL} \cdot d^{-\frac{1}{2}}\kappa^{-\frac{1}{2}}$

Table A.1. Optimal choices of leapfrog steps K and the step size η for the HMC algorithm for an (m, L, L_{H})-regular target distribution such that $L_{\mathrm{H}} = O(L^{\frac{3}{2}})$ used for the mixing time bounds in Corollary 6. Here c denotes a universal constant.

Proof of claim (A.74)**:** Recall that under the condition (A.51) (restated for reader's convenience)

$$\eta^2 \le \frac{1}{cL} \min \left\{ \frac{1}{K^2 d^{\frac{1}{2}}}, \frac{1}{K^2 d^{\frac{2}{3}}} \frac{L}{L_{\mathrm{H}}^{\frac{2}{3}}}, \frac{1}{K d^{\frac{1}{2}}}, \frac{1}{K^{\frac{2}{3}} d^{\frac{2}{3}} \kappa^{\frac{1}{3}} r(s)^{\frac{2}{3}}}, \frac{1}{K d^{\frac{1}{2}} \kappa^{\frac{1}{2}} r(s)}, \right.$$
$$\left. \frac{1}{K^{\frac{2}{3}} d} \frac{L}{L_{\mathrm{H}}^{\frac{2}{3}}}, \frac{1}{K^{\frac{4}{3}} d^{\frac{1}{2}} \kappa^{\frac{1}{2}} r(s)} \left(\frac{L}{L_{\mathrm{H}}^{\frac{2}{3}}} \right)^{\frac{1}{2}} \right\},$$

Corollary 3 guarantees that the HMC mixing time for the $\kappa^{\frac{d}{2}}$-warm initialization $\mu_\dagger = \mathcal{N}(x^\star, L^{-1}\mathbb{I}_d)$, is

$$\tau_2^{\mathrm{HMC}}(\epsilon; \mu_0) = O\left(d + \frac{\kappa}{K^2 \eta^2 L} \right),$$

where we have ignored logarithmic factors. In order to compare with MALA and other sampling methods, our goal is to optimize the number of gradient evaluations $\mathcal{G}_{\mathrm{eval}}$ taken by HMC to mix:

$$\mathcal{G}_{\mathrm{eval}} := K \cdot \tau_{\mathrm{TV}}^{\mathrm{HMC}}(\epsilon; \mu_0) = O\left(Kd + \frac{\kappa}{K\eta^2 L} \right). \tag{A.75}$$

Plugging in the condition on η stated above, we obtain

$$\mathcal{G}_{\text{eval}} \leq \max\Bigg\{ \underbrace{Kd}_{=:T_1}, \quad \underbrace{K \max\left(d^{\frac{1}{2}}\kappa, d^{\frac{2}{3}}\kappa\vartheta\right)}_{=:T_2}, \quad \underbrace{d^{\frac{1}{2}}\kappa^{\frac{3}{2}}}_{=:T_3}, \\ \underbrace{K^{-\frac{1}{3}}d^{\frac{2}{3}}\kappa^{\frac{4}{3}}}_{=:T_4}, \quad \underbrace{K^{-\frac{1}{3}}d\kappa\cdot\vartheta}_{=:T_5}, \quad \underbrace{K^{\frac{1}{3}}d^{\frac{1}{2}}\kappa^{\frac{3}{2}}\cdot\vartheta^{\frac{1}{2}}}_{=:T_6} \Bigg\} \tag{A.76}$$

where $\vartheta = L_{\text{H}}^{\frac{2}{3}}/L$. Note that this bound depends only on the relation between d, κ and the choice of K. We now summarize the source of all of these terms in our proofs:

- T_1: This term is attributed to the warmness of the initial distribution. The distribution $\mu_\dagger$ is $O(\kappa^d)$-warm. This term could be improved if we have a warmer initial distribution.
- T_2: This term appears in the proposal overlap bound from equation (3.23a) of Lemma 6 and more precisely, it comes from equation (3.31).
- T_3, T_4, T_5 and T_6: These terms pop-out from the accept-reject bound from equation (3.23b) of Lemma 6. More precisely, T_3 and T_4 are a consequence of the first three terms in equation (3.48), and T_5 and T_6 arise the last two terms in equation (3.48).

In Table A.2, we summarize how these six terms can be traded-off to derive the optimal parameter choices for Corollary 6. The effective bound on $\mathcal{G}_{\text{eval}}$-the number of gradient evaluations required by HMC to mix, is given by the largest of the six terms.

κ versus d	optimal K	T_1	T_2	T_3	T_4	T_5	T_6
		Kd	$Kd^{\frac{2}{3}}\kappa$	$d^{\frac{1}{2}}\kappa^{\frac{3}{2}}$	$K^{-\frac{1}{3}}d^{\frac{2}{3}}\kappa^{\frac{4}{3}}$	$K^{-\frac{1}{3}}d\kappa$	$K^{\frac{1}{3}}d^{\frac{1}{2}}\kappa^{\frac{3}{2}}$
$\kappa \in [1, d^{\frac{1}{3}})$	$K = \kappa^{\frac{3}{4}}$	$\mathbf{d\kappa^{\frac{3}{4}}}$	$d^{\frac{2}{3}}\kappa^{\frac{7}{4}}$	$d^{\frac{1}{2}}\kappa^{\frac{3}{2}}$	$d^{\frac{2}{3}}\kappa^{\frac{13}{12}}$	$\mathbf{d\kappa^{\frac{3}{4}}}$	$d^{\frac{1}{2}}\kappa^{\frac{7}{4}}$
$\kappa \in [d^{\frac{1}{3}}, d^{\frac{2}{3}}]$	$K = d^{\frac{1}{4}}$	$d^{\frac{5}{4}}$	$\mathbf{d^{\frac{11}{12}}\kappa}$	$d^{\frac{1}{2}}\kappa^{\frac{3}{2}}$	$d^{\frac{7}{12}}\kappa^{\frac{4}{3}}$	$\mathbf{d^{\frac{11}{12}}\kappa}$	$d^{\frac{7}{12}}\kappa^{\frac{3}{2}}$
$\kappa \in (d^{\frac{2}{3}}, d]$	$K = d^{\frac{3}{4}}\kappa^{-\frac{3}{4}}$	$d^{\frac{7}{4}}\kappa^{-\frac{3}{4}}$	$d^{\frac{19}{12}}\kappa^{\frac{1}{4}}$	$d^{\frac{1}{2}}\kappa^{\frac{3}{2}}$	$d^{\frac{5}{12}}\kappa^{\frac{19}{12}}$	$\mathbf{d^{\frac{3}{4}}\kappa^{\frac{5}{4}}}$	$\mathbf{d^{\frac{3}{4}}\kappa^{\frac{5}{4}}}$
$\kappa \in (d, \infty]$	$K = 1$	d	$d^{\frac{2}{3}}\kappa$	$\mathbf{d^{\frac{1}{2}}\kappa^{\frac{3}{2}}}$	$d^{\frac{2}{3}}\kappa^{\frac{4}{3}}$	$d\kappa$	$\mathbf{d^{\frac{1}{2}}\kappa^{\frac{3}{2}}}$

Table A.2. Trade-off between the six terms $T_i, i = 1, \dots 6$, from the bound (A.76) under the assumption $\vartheta = L_{\text{H}}^{2/3}/L \leq 1$. In the second column, we provide the optimal choice of K for the condition on κ stated in first column such that the maximum of the T_i's is smallest. For each row the dominant (maximum) term, and equivalently the effective bound on $\mathcal{G}_{\text{eval}}$ is displayed in bold (red).

Faster mixing time bounds

We now derive several mixing time bounds under additional assumptions: (a) when a warm start is available, and (b) the Hessian-Lipschitz constant is small.

Faster mixing time with warm start: When a better initialization with warmness $\varpi \leq O(e^{d^{\frac{2}{3}}\kappa})$ is available, and suppose that κ is much smaller than d. In such a case, the optimal choice turns out to be $K = d^{\frac{1}{4}}$ (instead of $\kappa^{\frac{3}{4}}$) which implies a bound of $O\left(d^{\frac{11}{12}}\kappa \log\left(\frac{1}{\epsilon}\right)\right)$ on $\mathcal{G}_{\text{eval}}$ (this bound was also stated in Table 3.1).

Faster mixing time with small L_{H}: Suppose in addition to warmness being not too large, $\varpi \leq O(e^{d^{\frac{2}{3}}\kappa})$, the Hessian-Lipschitz constant L_{H} is small enough $L_{\text{H}}^{\frac{2}{3}} \ll L$. In such a scenario, the terms T_5 and T_6 become negligible because of small L_{H} and T_1 is negligible because of small ϖ. The terms T_3 and T_4 remain unchanged, and the term T_2 changes slightly. More precisely, for the case $L_{\text{H}}^{\frac{2}{3}} \leq \frac{L}{d^{\frac{1}{2}}\kappa^{\frac{1}{2}}}$ we obtain a slightly modified trade-off for the terms in the (A.76) for $\mathcal{G}_{\text{eval}}$ (summarized in Table A.3). If κ is small too, then we obtain a mixing time bound of order $d^{\frac{5}{8}}$. Via this artificially constructed example, we wanted to demonstrate two things. First, faster convergence rates are possible to derive under additional assumptions directly from our results. Suitable adaptation of our proof techniques might provide a faster rate of mixing for Metropolized HMC under additional assumptions like infinity semi-norm regularity condition made in other works [128] (but we leave a detailed derivation for future work). Second, it also demonstrates the looseness of our proof techniques since we were unable to recover an $O(1)$ mixing time bound for sampling from a Gaussian target.

κ versus d	K optimal choice	T_1	T_2	T_3	T_4	T_5	T_6
		-	$Kd^{\frac{1}{2}}\kappa$	$d^{\frac{1}{2}}\kappa^{\frac{3}{2}}$	$K^{-\frac{1}{3}}d^{\frac{2}{3}}\kappa^{\frac{4}{3}}$	-	-
$\kappa \in (0, d^{\frac{1}{2}})$	$K = d^{\frac{1}{8}}\kappa^{\frac{1}{4}}$	-	$\mathbf{d^{\frac{5}{8}}\kappa^{\frac{5}{4}}}$	$d^{\frac{1}{2}}\kappa^{\frac{3}{2}}$	$\mathbf{d^{\frac{5}{8}}\kappa^{\frac{5}{4}}}$	-	-

Table A.3. Six terms in the HMC number of gradient evaluations bound under small hessian-Lipschitz constant and very warm start. The dominant term is highlighted in red.

Linearly transformed HMC (effect of mass function): In practice, it is often beneficial to apply linear transformations in HMC (cf. Section 4 [145]). At a high level, such a transformation can improve the conditioning of the problem and help HMC mix faster. For the target distribution Π^* with density proportional to e^{-f}, we can define a new distribution Π_h with density e^{-h} (up to normalization) such that $h(x) = f(M^{-\frac{1}{2}}x)$ where $M \in \mathbb{R}^{d\times d}$ is an invertible matrix. Then for a random sample $\tilde{q} \sim \Pi_h$, the

distribution of $M^{\frac{1}{2}}\tilde{q}$ is Π^*. When the new distribution h has a better condition number κ_h than the condition number κ of f, we can use HMC to draw approximate sample from Π_h and then transform the samples using the matrix M. Clearly the bound from Corollary 6 guarantees that when κ_h is much smaller than κ, HMC on the new target Π_h would mix much faster than the HMC chain on Π^*. This transformation is equivalent to the HMC algorithm with modified kinetic energy

$$\frac{dq_t}{dt} = M^{-1}p_t \quad \text{and} \quad \frac{dp_t}{dt} = -\nabla f(q_t),$$

which is easier to implement in practice. For a detailed discussion of this implementation, we refer the readers to the paper by Neal [145].

A.4.2 Comparison with guarantees for unadjusted versions of HMC

In this appendix, we compare our results with mixing time guarantees results on unadjusted and ODE solver based HMC chains. We summarize the number of gradient evaluations needed for Metropolized HMC to mix and those for other existing sampling results in Table A.4. Note that all the results summarized here are the best upper bounds in the literature for log-concave sampling. We present the results for a (L, L_{H}, m)-regular target distribution. We remark that all methods presented in Table A.4 requires the regularity assumptions (3.6a) and (3.6b), even though some do not require assumption (3.6c).

Two remarks are in order. First, the error metric for the guarantees in the works [128, 39, 108] is 1-Wasserstein distance, while our results make use of $\mathcal{L}_2$ or TV distance. As a result, a direct comparison between these results is not possible although we provide an indirect comparison below. Second, the previous guarantees have a polynomial dependence on the inverse of error-tolerance $1/\epsilon$. In contrast, our results for MALA and Metropolized HMC have a logarithmic dependence $\log(1/\epsilon)$. For a well-conditioned target, i.e., when κ is a constant, all prior results have a better dependence on d when compared to our bounds.

Logarithmic vs polynomial dependence on $1/\epsilon$: We now provide an indirect comparison, between prior guarantees based on Wasserstein distance and our results based on TV-distance, for estimating expectations of Lipschitz-functions on bounded domains. MCMC algorithms are used to estimate expectations of certain functions of interest. Given an arbitrary function g and an MCMC algorithm, one of the ways to estimate $\Pi^*(g) := \mathbb{E}_{X\sim\Pi^*}[g(X)]$ is to use the k-th iterate from N independent runs of the chain. Let $X_i^{(k)}$ for $i = 1, \ldots, N$ denote the N i.i.d. samples at the k-th iteration of the chain and let μ_k denote the distribution of $X_i^{(k)}$, namely the distribution of the chain after k iterations. Then for the estimate $\widehat{\Pi}_k(g) := \frac{1}{N}\sum_{i=1}^N g(X_i^{(k)})$, the estimation

Sampling algorithm	#Grad. evals
$^{\ddagger,\diamond}$Unadjusted HMC with leapfrog integrator [128]	$d^{\frac{1}{4}}\kappa^{\frac{11}{4}} \cdot \frac{1}{\epsilon^{1/2}}$
‡Underdamped Langevin [39]	$d^{\frac{1}{2}}\kappa^{2} \cdot \frac{1}{\epsilon}$
‡HMC with ODE solver, Thm 1.6 in [108]	$d^{\frac{1}{2}}\kappa^{\frac{7}{4}} \cdot \frac{1}{\epsilon}$
*MALA [58][this chapter]	$\max\left\{d\kappa, d^{\frac{1}{2}}\kappa^{\frac{3}{2}}\right\} \cdot \log\frac{1}{\epsilon}$
*Metropolized HMC with leapfrog integrator [this chapter]	$\max\left\{d\kappa^{\frac{3}{4}}, d^{\frac{11}{12}}\kappa, d^{\frac{3}{4}}\kappa^{\frac{5}{4}}, d^{\frac{1}{2}}\kappa^{\frac{3}{2}}\right\} \cdot \log\frac{1}{\epsilon}$

Table A.4. Summary of the number of gradient evaluations needed for the sampling algorithms to converge to a (m, L, L_{H})-regular target distribution with $L_{\mathrm{H}} = O(L^{\frac{3}{2}})$ within ϵ error from the target distribution (in total-variation distance* or 1-Wasserstein distance‡) (and $\diamond$ certain additional regularity conditions for the result by Mangoubi et al. [128]). Note that the unadjusted algorithms suffer from an exponentially worse dependency on ϵ when compared to the Metropolis adjusted chains. For MALA, results by Dwivedi et al. [58] had an extra d factor which is sharpened in Theorem 4.

error can be decomposed as

$$\begin{aligned}\Pi^*(g) - \widehat{\Pi}_k(g) &= \int_{\mathbb{R}^d} g(x)\pi^*(x)dx - \frac{1}{N}\sum_{i=1}^{N} g(X_i^{(k)}) \\ &= \underbrace{\int_{\mathbb{R}^d} g(x)\left[\pi^*(x) - \mu_k(x)\right]dx}_{=:J_1 \text{ (Approximation bias)}} + \underbrace{\mathbb{E}_{\mu_k}\left[g(X)\right] - \frac{1}{N}\sum_{i=1}^{N} g(X_i^{(k)})}_{=:J_2 \text{ (Finite sample error)}}. \end{aligned} \tag{A.77}$$

To compare different prior works, we assume that $\mathrm{Var}_{\mu_k}\left[g(X_1)\right]$ is bounded and thereby that the finite sample error J_2 is negligible for large enough N.[5] It remains to bound the error J_1 which can be done in two different ways depending on the error-metric used to provide mixing time guarantees for the Markov chain.

If the function g is ω-Lipschitz and k is chosen such that $\mathcal{W}_1(\Pi^*, \mu_k) \leq \epsilon$, then we have $J_1 \leq \omega\epsilon =: J_{\mathrm{Wass}}$. On the other hand, if the function g is bounded by B, and k is chosen such that $d_{\mathrm{TV}}\left(\Pi^*, \mu_k\right) \leq \epsilon$, then we obtain the bound $J_1 \leq B\epsilon =: J_{\mathrm{TV}}$. We make use of these two facts to compare the number of gradient evaluations needed by unadjusted HMC or ODE solved based HMC and Metropolized HMC. Consider an ω-Lipschitz function g with support on a ball of radius R. Note that this function is uniformly bounded by $B = \omega R$. Now in order to to ensure that $J_1 \leq \delta$ (some

[5] Moreover, this error should be usually similar across different sampling algorithms since several algorithms are designed in a manner agnostic to a particular function g.

user-specified small threshold), the choice of ϵ in the two cases (Wasserstein and TV distance) would be different leading to different number of gradient evaluations required by the two chains. More precisely, we have

$$J_1 \leq J_{\text{Wass}} = \omega\epsilon \leq \delta \quad \Longrightarrow \quad \epsilon_{\text{wass}} = \frac{\delta}{\omega} \quad \text{and}$$
$$J_1 \leq J_{\text{TV}} = B\epsilon = \omega R\epsilon \leq \delta \quad \Longrightarrow \quad \epsilon_{\text{TV}} = \frac{\delta}{\omega R}.$$

To simplify the discussion, we consider well-conditioned (constant κ) strongly log-concave distributions such that most of the mass is concentrated on a ball of radius $O(\sqrt{d})$ (cf. Appendix A.2.1) and consider $R = \sqrt{d}$. Then plugging the error-tolerances from the display above in Table A.4, we obtain that the number of gradient evaluations $\mathcal{G}_{\text{MC}}$ for different chains[6] would scale as

$$\mathcal{G}_{\text{unadj.-HMC}} \leq O(\sqrt{\frac{d\omega}{\delta}}), \quad \mathcal{G}_{\text{ODE-HMC}} \leq O(\frac{\omega\sqrt{d}}{\delta}), \text{and} \quad \mathcal{G}_{\text{Metro.-HMC}} \leq O(d \log \frac{\omega\sqrt{d}}{\delta})$$

Clearly, depending on ω and the threshold δ, different chains would have better guarantees. When ω is large or δ is small, our results ensure the superiority of Metropolized-HMC over other versions. For example, higher-order moments can be functions of interest, i.e., $g(x) = \|x\|^{1+\nu}$ for which the Lipschitz-constant $\omega = O(d^\nu)$ scales with d. For this function, we obtain the bounds:

$$\mathcal{G}_{\text{unadj.-HMC}} \leq O(\frac{d^{\frac{1+\nu}{2}}}{\sqrt{\delta}}), \quad \mathcal{G}_{\text{ODE-HMC}} \leq O(\frac{d^{\frac{1}{2}+\nu}}{\delta}), \quad \text{and} \quad \mathcal{G}_{\text{Metro.-HMC}} \leq O(d(1+\nu) \log \frac{d}{\delta})$$

and thus Metropolized HMC takes fewer gradient evaluations than ODE-based HMC for $\nu > 1/2$ and unadjusted HMC for $\nu > 1$ (to ensure $J_1 \leq \delta$ (A.77)). We remark that the bounds for unadjusted-HMC require additional regularity conditions. From this informal comparison, we demonstrate that both the dimension dependency d and error dependency ϵ should be accounted for comparing unadjusted algorithms and Metropolized algorithms. Especially for estimating high-order moments, Metropolized algorithms with $\log(\frac{1}{\epsilon})$ dependency will be advantageous.

[6]The results for other HMCs often assume (different) additional conditions so that a direct comparison should be taken with a fine grain of salt.

Appendix B

Technical proofs for Vaidya and John walks

B.1 Auxiliary results for the Vaidya walk

In this appendix, we first summarize a few notations used in the proofs related to Theorem 5, and collect the auxiliary results for the later proofs.

B.1.1 Notation

We begin with introducing the notation. Recall $A \in \mathbb{R}^{n\times d}$ is a matrix with $a_i^\top$ as its i-th row. For any positive integer p and any vector $v = (v_1, \ldots, v_p)^\top$, $\mathrm{diag}(v) = \mathrm{diag}(v_1, \ldots, v_p)$ denotes a $p \times p$ diagonal matrix with the i-th diagonal entry equal to v_i. Recall the definition of S_x:

$$S_x = \mathrm{diag}\left(s_{x,1}, \ldots, s_{x,n}\right) \text{ where } s_{x,i} = b_i - a_i^\top x \text{ for each } i \in [n]. \tag{B.1}$$

Furthermore, define $A_x = S_x^{-1}A$ for all $x \in \mathrm{int}\left(\mathcal{K}\right)$, and let Υ_x denote the projection matrix for the column space of A_x, i.e.,

$$\Upsilon_x := A_x(A_x^\top A_x)^{-1}A_x^\top = A_x\nabla^2\mathcal{F}_x^{-1}A_x^\top. \tag{B.2}$$

Note that for the scores σ_x (4.6b), we have $\sigma_{x,i} = (\Upsilon_x)_{ii}$ for each $i \in [n]$. Let Σ_x be an $n \times n$ diagonal matrix defined as

$$\Sigma_x = \mathrm{diag}\left(\sigma_{x,1}, \ldots, \sigma_{x,n}\right). \tag{B.3}$$

Let $\sigma_{x,i,j} := (\Upsilon_x)_{ij}$, and let $\Upsilon_x^{(2)}$ denote the Hadamard product of Υ_x with itself, i.e.,

$$(\Upsilon_x^{(2)})_{ij} = \sigma_{x,i,j}^2 = \frac{\left(a_i^\top\nabla^2\mathcal{F}_x^{-1}a_j\right)^2}{s_{x,i}^2 s_{x,j}^2} \quad \text{for all } i, j \in [n]. \tag{B.4}$$

Using the shorthand $\theta_x := \theta_{V_x}$, we define

$$\Theta_x := \operatorname{diag}(\theta_{x,1}, \ldots, \theta_{x,m}) \quad \text{where } \theta_{x,i} = \frac{a_i^\top V_x^{-1} a_i}{s_{x,i}^2} \quad \text{for } i \in [n], \text{ and}$$

$$\Xi_x := (\theta_{x,i,j}^2) \quad \text{where } \theta_{x,i,j}^2 = \frac{\left(a_i^\top V_x^{-1} a_j\right)^2}{s_{x,i}^2 s_{x,j}^2} \quad \text{for } i, j \in [n].$$

In our new notation, we can re-write the Vaidya matrix V_x defined in equation (4.6a) as $V_x = A_x^\top (\Sigma_x + \beta_{\mathrm{V}} \mathbb{I}) A_x$, where $\beta_{\mathrm{V}} = d/n$.

B.1.2 Basic Properties

We begin by summarizing some key properties of various terms involved in our analysis.

Lemma 23. *For any vector $x \in \operatorname{int}(\mathcal{K})$, the following properties hold:*

(a) $\sigma_{x,i} = \sum_{j=1}^n \sigma_{x,i,j}^2 = \sum_{j,k=1}^n \sigma_{x,i,j}\sigma_{x,j,k}\sigma_{x,k,i}$ *for each* $i \in [n]$,

(b) $\Sigma_x \succeq \Upsilon_x^{(2)}$,

(c) $\sum_{i=1}^n \theta_{x,i} (\sigma_{x,i} + \beta_V) = d$,

(d) $\forall i \in [n],\ \theta_{x,i} = \sum_{j=1}^n (\sigma_{x,j} + \beta_V) \theta_{x,i,j}^2$, *for each* $i \in [n]$,

(e) $\theta_x^\top (\Sigma_x + \beta_V \mathbb{I}) \theta_x = \sum_{i=1}^n \theta_{x,i}^2 (\sigma_{x,i} + \beta_V) \leq \sqrt{nd}$, *and*

(f) $\beta_V \nabla^2 \mathcal{F}_x \preceq V_x \preceq (1 + \beta_V) \nabla^2 \mathcal{F}_x$.

where $\beta_V = d/n$ was defined in equation (4.6b).

Proof. We prove each property separately.

Part (a): Using $\mathbb{I}_d = \nabla^2 \mathcal{F}_x (\nabla^2 \mathcal{F}_x)^{-1}$, we find that

$$\begin{aligned}
\sigma_{x,i} &= \frac{a_i^\top (\nabla^2 \mathcal{F}_x)^{-1} \nabla^2 \mathcal{F}_x (\nabla^2 \mathcal{F}_x)^{-1} a_i}{s_{x,i}^2} \\
&= \frac{a_i^\top (\nabla^2 \mathcal{F}_x)^{-1} \nabla^2 \sum_{j=1}^n \frac{a_j^\top a_j}{s_{x,j}^2} (\nabla^2 \mathcal{F}_x)^{-1} a_i}{s_{x,i}^2} \\
&= \sum_{i,j=1}^n \sigma_{x,i,j}^2.
\end{aligned}$$

Applying a similar trick twice and performing some algebra, we obtain

$$\sigma_{x,i} = \frac{a_i^\top (\nabla^2 \mathcal{F}_x)^{-1} \nabla^2 \mathcal{F}_x (\nabla^2 \mathcal{F}_x)^{-1} \nabla^2 \mathcal{F}_x (\nabla^2 \mathcal{F}_x)^{-1} a_i}{s_{x,i}^2} = \sum_{i,j,k=1}^n \sigma_{x,i,j}\sigma_{x,j,k}\sigma_{x,k,i}.$$

Part (b): From part (a), we have that $\Sigma_x - \Upsilon_x^{(2)}$ is a symmetric and diagonally dominant matrix with non-negative entries on the diagonal. Applying Gershgorin's theorem [15, 80], we conclude that it is PSD.

Part (c): Since $\mathrm{trace}(AB) = \mathrm{trace}(BA)$, we have

$$\sum_{i=1}^{n} \theta_{x,i} \left(\sigma_{x,i} + \beta_{\mathrm{V}}\right) = \mathrm{trace}\left(V_x^{-1} \sum_{i=1}^{n} \left(\sigma_{x,i} + \beta_{\mathrm{V}}\right) \frac{a_i a_i^\top}{s_{x,i}^2}\right) = \mathrm{trace}\left(\mathbb{I}_d\right) = d.$$

Part (d): An argument similar to part (a) implies that

$$\theta_{x,i} = \frac{a_i^\top V_x^{-1} V_x V_x^{-1} a_i}{s_{x,i}^2} = \frac{a_i^\top V_x^{-1} \sum_{j=1}^{n} \left(\sigma_{x,i} + \beta_{\mathrm{V}}\right) \frac{a_j^\top a_j}{s_{x,j}^2} V_x^{-1} a_i}{s_{x,i}^2} = \sum_{i,j=1}^{n} \left(\sigma_{x,i} + \beta_{\mathrm{V}}\right) \theta_{x,i,j}^2.$$

Part (e): Using part (c) and Lemma 11(c) yields the claim.

Part (f): The left inequality is by the definition of V_x. The right inequality uses the fact that $\Sigma_x \preceq \mathbb{I}_d$. □

We now prove an important result that relates the *slackness* s_x and s_y at two points, in terms of $\|x - y\|_x$.

Lemma 24. *For all $x, y \in \mathrm{int}\left(\mathcal{K}\right)$, we have*

$$\left|1 - \frac{s_{y,i}}{s_{x,i}}\right| \leq \left(\frac{n}{d}\right)^{\frac{1}{4}} \|x - y\|_x \quad \textit{for each } i \in [n].$$

Proof. For any pair $x, y \in \mathrm{int}\left(\mathcal{K}\right)$ and index $i \in [n]$, we have

$$\begin{aligned}
\left(a_i^\top \left(x - y\right)\right)^2 = \left((V_x^{-\frac{1}{2}} a_i)^\top V_x^{\frac{1}{2}} (x - y)\right)^2 &\overset{(i)}{\leq} \|V_x^{-\frac{1}{2}} a_i\|_2^2 \, \|V_x^{\frac{1}{2}} (x - y)\|_2^2 \\
&= a_i^T V_x^{-1} a_i \;\; \|x - y\|_x^2 \\
&= \theta_{x,i} s_{x,i}^2 \;\; \|x - y\|_x^2 \\
&\overset{(ii)}{\leq} \sqrt{\frac{n}{d}} s_{x,i}^2 \;\; \|x - y\|_x^2,
\end{aligned}$$

where step (i) follows from the Cauchy-Schwarz inequality, and step (ii) uses the bound $\theta_{x,i}$ from Lemma 11(c). Noting the fact that $a_i^\top (x - y) = s_{y,i} - s_{x,i}$, the claim follows after simple algebra. □

B.2 Proof of Lemma 13

In this appendix section, we prove Lemma 13 using results from the previous appendix. As a direct consequence of Lemma 24, we find that

$$\left|1 - \frac{s_{y,i}}{s_{x,i}}\right| \leq \frac{t}{\sqrt{d}}, \quad \text{for any } x, y \in \operatorname{int}(\mathcal{K}) \text{ such that } \|x - y\|_x \leq \frac{t}{(nd)^{1/4}}.$$

The Hessian $\nabla^2 \mathcal{F}_y$ is thus sandwiched in terms of the Hessian $\nabla^2 \mathcal{F}_x$ as

$$\left(1 - \frac{t}{\sqrt{d}}\right)^2 \nabla^2 \mathcal{F}_x \preceq \nabla^2 \mathcal{F}_y \preceq \left(1 + \frac{t}{\sqrt{d}}\right)^2 \nabla^2 \mathcal{F}_x.$$

By the definition of $\sigma_{x,i}$ and $\sigma_{y,i}$, we have

$$\frac{\left(1 - \frac{t}{\sqrt{d}}\right)^2}{\left(1 + \frac{t}{\sqrt{d}}\right)^2} \sigma_{x,i} \leq \sigma_{y,i} \leq \frac{\left(1 + \frac{t}{\sqrt{d}}\right)^2}{\left(1 - \frac{t}{\sqrt{d}}\right)^2} \sigma_{x,i} \quad \text{for all } i \in [n]. \tag{B.5}$$

Consequently, we find that

$$\frac{\left(1 - \frac{t}{\sqrt{d}}\right)^2}{\left(1 + \frac{t}{\sqrt{d}}\right)^4} V_x \preceq V_y \preceq \frac{\left(1 + \frac{t}{\sqrt{d}}\right)^2}{\left(1 - \frac{t}{\sqrt{d}}\right)^4} V_x.$$

Note that

$$\frac{(1-\omega)^2}{(1+\omega)^4} \geq 1 - 8\omega \quad \text{and} \quad \frac{(1+\omega)^2}{(1-\omega)^4} \leq 1 + 8\omega \quad \text{for any } \omega \in \left[0, \tfrac{1}{12}\right].$$

Applying this sandwiching pair of inequalities with $\omega = t/\sqrt{d}$ yields the claim.

B.3 Proof of Lemma 14

We begin by defining

$$\varphi_{x,i} := \frac{\sigma_{x,i} + \beta_{\mathrm{V}}}{s_{x,i}^2} \text{ for } i \in [n], \quad \text{and} \quad \Psi_x := \frac{1}{2} \log\det V_x, \quad \text{for all } x \in \operatorname{int}(\mathcal{K}). \tag{B.6}$$

Further, for any two points x and z, let $\overline{xz}$ denote the set of points on the line segment joining x and z. The proof of Lemma 14 is based on a Taylor series expansion, and so requires careful handling of σ, φ, Ψ and their derivatives. At a high level, the proof involves the following steps: (1) perform a Taylor series expansion around x and along the line segment $\overline{xz}$; (2) transfer the bounds of terms involving some point $y \in \overline{xz}$ to terms involving only x and z; and then (3) use concentration of Gaussian polynomials to obtain high probability bounds.

B.3.1 Auxiliary results for the proof of Lemma 14

We now introduce some auxiliary results involved in these three steps. The following lemma provides expressions for gradients of σ, φ and Ψ and bounds for directional Hessian of φ and Ψ. Let $e_i \in \mathbb{R}^d$ denote a vector with 1 in the i-th position and 0 otherwise. For any $h \in \mathbb{R}^d$ and $x \in \operatorname{int}(\mathcal{K})$, define $\eta_{x,h,i} = \eta_{x,i} := a_i^\top h / s_{x,i}$ for each $i \in [n]$.

Lemma 25. *The following relations hold;*

(a) Gradient of σ: $\nabla \sigma_{x,i} = 2A_x^\top (\Sigma_x - \Upsilon_x^{(2)}) e_i$ *for each* $i \in [n]$.

(b) Gradient of φ: $\nabla \varphi_{x,i} = \dfrac{2}{s_{x,i}^2} A_x^\top \left[2\Sigma_x + \beta_V \mathbb{I} - \Upsilon_x^{(2)}\right] e_i$ *for each* $i \in [n]$;

(c) Gradient of Ψ: $\nabla \Psi_x = A_x^\top \left(2\,\Sigma_x + \beta_V \mathbb{I} - \Upsilon_x^{(2)}\right) \theta_x$;

(d) Bound on $\nabla^2 \varphi$: $s_{x,i}^2 \left|\frac{1}{2} h^\top \nabla^2 \varphi_{x,i} h\right| \le 14\,(\sigma_{x,i} + \beta_V)\,\eta_{x,i}^2 + 11 \sum_{j=1}^n \sigma_{x,i,j}^2 \eta_{x,j}^2$ *for* $i \in [n]$;

(e) Bound on $\nabla^2 \Psi$: $\left|\frac{1}{2} h^\top (\nabla^2 \Psi_x)\, h\right| \le 13 \sum_{i=1}^n (\sigma_{x,i} + \beta_V)\,\theta_{x,i} \eta_{x,i}^2 + \frac{17}{2} \sum_{i,j=1}^n \sigma_{x,i,j}^2 \theta_{x,i} \eta_{x,j}^2$.

See Section B.3.6 for the proof of this claim.

The following lemma that shows that for a random variable $z \sim \mathcal{P}_x$, the slackness $s_{z,i}$ is close to $s_{x,i}$ with high probability.

Lemma 26. *For any $\epsilon \in (0, 1/4], r \in (0,1)$ and $x \in \operatorname{int}(\mathcal{K})$, we have*

$$\mathbb{P}_{z \sim \mathcal{P}_x}\left[\forall i \in [n], \forall v \in \overline{xz},\ \frac{s_{x,i}}{s_{v,i}} \in (1 - r\,(1+\delta)\,, 1 + r\,(1+\delta))\right] \ge 1 - \epsilon/4,$$

where $\delta = \sqrt{\frac{2\log(4/\epsilon)}{d}}$. Thus for any $d \ge 1$ and $r \le 1/\left[20\left(1 + \sqrt{2\log\left(\frac{4}{\epsilon}\right)}\right)\right]$, we have

$$\mathbb{P}_{z \sim \mathcal{P}_x}\left[\forall i \in [n], \forall v \in \overline{xz},\ \frac{s_{x,i}}{s_{v,i}} \in (0.95, 1.05)\right] \ge 1 - \epsilon/4.$$

See Section B.3.4 for the proof which is based on combining the bound on $\frac{s_{x,i}}{s_{v,i}}$ from Lemma 24 with standard Gaussian tail bounds.

This result comes in handy for transferring bounds for different expressions in Taylor expansion involving an arbitrary y on $\overline{xz}$ to bounds on terms involving simply x. The proof follows from Lemma 24 and a simple application of the standard Gaussian tail bounds and is thereby omitted. For brevity, we define the shorthand

$$\hat{a}_{x,i} = \frac{1}{s_{x,i}} V_x^{-1/2} a_i \quad \text{for each } i \in [n]. \tag{B.7}$$

In the following lemma, we state some tail bounds for particular Gaussian polynomials that arise in our analysis.

Lemma 27. *For any $\epsilon \in (0, 1/15]$, define $\chi_k = (2e/k \cdot \log(4/\epsilon))^{k/2}$ for $k = 2, 3$ and 4. Then for $\xi \sim \mathcal{N}(0, \mathbb{I}_d)$ and any $x \in \operatorname{int}(\mathcal{K})$ the following high probability bounds hold:*

$$\mathbb{P}\left[\sum_{i=1}^{n} (\sigma_{x,i} + \beta_V)\left(\hat{a}_{x,i}^\top \xi\right)^2 \leq \chi_2 \sqrt{3} d\right] \geq 1 - \frac{\epsilon}{4}, \tag{B.8a}$$

$$\mathbb{P}\left[\left|\sum_{i=1}^{n} (\sigma_{x,i} + \beta_V)\left(\hat{a}_{x,i}^\top \xi\right)^3\right| \leq \chi_3 \sqrt{15}\,(nd)^{1/4}\right] \geq 1 - \frac{\epsilon}{4}, \tag{B.8b}$$

$$\mathbb{P}\left[\left|\sum_{i,j=1}^{n} \sigma_{x,i,j}^2 \left(\left(\frac{\hat{a}_{x,i} + \hat{a}_{x,j}}{2}\right)^\top \xi\right)^3\right| \leq \chi_3 \sqrt{15}\,(nd)^{1/4}\right] \geq 1 - \frac{\epsilon}{4}, \tag{B.8c}$$

$$\mathbb{P}\left[\sum_{i=1}^{n} (\sigma_{x,i} + \beta_V)\left(\hat{a}_{x,i}^\top \xi\right)^4 \leq \chi_4 \sqrt{105}\,(nd)^{1/2}\right] \geq 1 - \frac{\epsilon}{4}. \tag{B.8d}$$

See Section B.3.5 for the proof of these claims.

Now we summarize the final ingredients needed for our proofs. Recall that the Gaussian proposal z is related to the current state x via the equation

$$z \overset{d}{=} x + \frac{r}{(nd)^{1/4}} V_x^{-1/2} \xi, \tag{B.9}$$

where $\xi \sim \mathcal{N}(0, \mathbb{I}_d)$. We also use the following elementary inequalities:

Cauchy-Schwarz inequality:	$\lvert u^\top v \rvert \leq \lVert u \rVert_2 \lVert v \rVert_2$	(C-S)
AM-GM inequality:	$\nu\kappa \leq \frac{1}{2}(\nu^2 + \kappa^2)$.	(AM-GM)
Sum of squares inequality:	$\frac{1}{2}\lVert a + b \rVert_2^2 \leq \lVert a \rVert_2^2 + \lVert b \rVert_2^2$,	(SSI)

Note that the sum-of-squares inequality is simply a vectorized version of the AM-GM inequality. With these tools, we turn to the proof of Lemma 14. We split our analysis into parts.

B.3.2 Proof of claim (4.29a)

Using the second degree Taylor expansion, we have

$$\Psi_z - \Psi_x = (z - x)^\top \nabla \Psi_x + \frac{1}{2}(z - x)^\top \nabla^2 \Psi_y (z - x), \quad \text{for some } y \in \overline{xz}.$$

We claim that for $r \leq f(\epsilon)$, we have

$$\mathbb{P}_z\left[(z - x)^\top \nabla \Psi_x \geq -\epsilon/2\right] \geq 1 - \epsilon/2, \quad \text{and} \tag{B.10a}$$

$$\mathbb{P}_z\left[\frac{1}{2}(z - x) \nabla^2 \Psi_y (z - x) \geq -\epsilon/2\right] \geq 1 - \epsilon/2. \tag{B.10b}$$

Note that the claim (4.29a) is a consequence of these two auxiliary claims, which we now prove.

Proof of bound (B.10a)

Equation (B.9) implies that $(z-x)^\top \nabla\Psi_x \sim \mathcal{N}\left(0, \frac{r^2}{\sqrt{nd}}\nabla\Psi_x^\top V_x^{-1}\nabla\Psi_x\right)$. We claim that

$$\nabla\Psi_x^\top V_x^{-1}\nabla\Psi_x \leq 9\sqrt{nd} \quad \text{for all } x \in \operatorname{int}(\mathcal{K}). \tag{B.11}$$

We prove this inequality at the end of this subsection. Taking it as given for now, let $\xi' \sim \mathcal{N}(0, 9r^2)$. Then using inequality (B.11) and a standard Gaussian tail bound, we find that

$$\mathbb{P}\left[(z-x)^\top \nabla\Psi_x \geq -\omega\right] \geq \mathbb{P}\left[\xi' \geq -\omega\right] \geq 1 - \exp(-\omega^2/(18r^2)), \quad \text{valid for all } \omega \geq 0.$$

Setting $\omega = \epsilon/2$ and noting that $r \leq \frac{\epsilon}{\sqrt{18\log(2/\epsilon)}}$ completes the claim.

Proof of bound (B.10b)

Let $\eta_{x,i} = \frac{a_i^\top(z-x)}{s_{x,i}} = \frac{r}{(mn)^{\frac{1}{4}}}\hat{a}_{x,i}^\top\xi$. Using Lemma 25(e), we have

$$\begin{aligned}\left|\frac{1}{2}(z-x)^\top\nabla^2\Psi_y(z-x)\right| &\leq 13\sum_{i=1}^n(\sigma_{y,i}+\beta_{\mathrm{V}})\theta_{y,i}\frac{s_{x,i}^2}{s_{y,i}^2}\eta_{x,i}^2 + \frac{17}{2}\sum_{i,j=1}^n\sigma_{y,i,j}^2\theta_{y,i}\frac{s_{x,j}^2}{s_{y,j}^2}\eta_{x,j}^2 \\ &\leq \frac{43}{2}\sqrt{\frac{n}{d}}\sum_{i=1}^n(\sigma_{x,i}+\beta_{\mathrm{V}})\frac{(\sigma_{y,i}+\beta_{\mathrm{V}})}{(\sigma_{x,i}+\beta_{\mathrm{V}})}\frac{s_{x,i}^2}{s_{y,i}^2}\eta_{x,i}^2. \end{aligned}\tag{B.12}$$

The last inequality comes from Lemma 11(c) and Lemma 23(a). Setting $\tau = 1.05$, we define the events $\mathcal{E}_1$ and $\mathcal{E}_2$ as follows:

$$\mathcal{E}_1 = \left\{\forall i \in [n], \frac{s_{x,i}}{s_{y,i}} \in [2-\tau, \tau]\right\}, \quad \text{and} \tag{B.13a}$$

$$\mathcal{E}_2 = \left\{\forall i \in [n], \frac{\sigma_{x,i}}{\sigma_{y,i}} \in \left[0, \frac{\tau^2}{(2-\tau)^2}\right]\right\}. \tag{B.13b}$$

It is straightforward to see that $\mathcal{E}_1 \subseteq \mathcal{E}_2$ following a similar argument we used to obtain equation (B.5) in the proof of Lemma 13. Since $r \leq 1/\left[20\left(1+\sqrt{2}\log^{1/2}\left(\frac{4}{\epsilon}\right)\right)\right]$, Lemma 26 implies that $\mathbb{P}[\mathcal{E}_1] \geq 1-\epsilon/4$ whence $\mathbb{P}[\mathcal{E}_2] \geq 1-\epsilon/4$. Using these high probability bounds and the setting $\tau = 1.05$, we obtain that with probability at least $1-\epsilon/4$

$$\sqrt{\frac{n}{d}}\sum_{i=1}^n(\sigma_{x,i}+\beta_{\mathrm{V}})\frac{(\sigma_{y,i}+\beta_{\mathrm{V}})}{(\sigma_{x,i}+\beta_{\mathrm{V}})}\frac{s_{x,i}^2}{s_{y,i}^2}\eta_{x,i}^2 \tag{B.14}$$

$$\leq 2\sqrt{\frac{n}{d}}\sum_{i=1}^n(\sigma_{x,i}+\beta_{\mathrm{V}})\eta_{x,i}^2 \tag{B.15}$$

$$= \frac{2r^2}{d}\sum_{i=1}^n(\sigma_{x,i}+\beta_{\mathrm{V}})(\hat{a}_{x,i}^\top\xi)^2. \tag{B.16}$$

Applying the high probability bound Lemma 27 (B.8a) and the condition

$$r \leq \sqrt{\frac{\epsilon}{86\sqrt{3}\chi_2}}, \tag{B.17}$$

we obtain that with probability at least $1 - \epsilon/2$,

$$\frac{1}{2}(z-x)^\top \nabla^2 \Psi_y (z-x) \geq -\epsilon/2,$$

as claimed.

Proof of bound (B.11)

We now return to prove our earlier inequality (B.11). Using the expression for the gradient $\nabla \Psi_x$ from Lemma 25(c), we have that for any vector $u \in \mathbb{R}^n$

$$\begin{aligned} u^\top \nabla \Psi_x \nabla \Psi_x^\top u &= \left\langle u, A_x^\top \left(2\Sigma_x - \Upsilon_x^{(2)} + \beta_{\mathrm{V}} \mathbb{I}\right) \theta_x \right\rangle^2 \\ &= \left\langle A_x u, \left(2\Sigma_x - \Upsilon_x^{(2)} + \beta_{\mathrm{V}} \mathbb{I}\right) \theta_x \right\rangle^2 \\ &= \left\langle (\Sigma_x + \beta_{\mathrm{V}} \mathbb{I})^{\frac{1}{2}} A_x u, (\Sigma_x + \beta_{\mathrm{V}} \mathbb{I})^{-1/2} \left(2\Sigma_x - \Upsilon_x^{(2)} + \beta_{\mathrm{V}} \mathbb{I}\right) \theta_x \right\rangle^2 \\ &\leq u^\top V_x u \cdot \theta_x^\top \left(2\Sigma_x - \Upsilon_x^{(2)} + \beta_{\mathrm{V}} \mathbb{I}\right) (\Sigma_x + \beta_{\mathrm{V}} \mathbb{I})^{-1} \left(2\Sigma_x - \Upsilon_x^{(2)} + \beta_{\mathrm{V}} \mathbb{I}\right) \theta_x \end{aligned} \tag{B.18}$$

where the last step follows from the Cauchy-Schwarz inequality. As a consequence of Lemma 23(b), the matrix $\Sigma_x - \Upsilon_x^{(2)}$ is PSD. Thus, we have

$$0 \preceq 2\Sigma_x - \Upsilon_x^{(2)} + \beta_{\mathrm{V}} \mathbb{I} \preceq 3(\Sigma_x + \beta_{\mathrm{V}} \mathbb{I}).$$

Consequently, we find that

$$0 \preceq \underbrace{(3\Sigma_x + 3\beta_{\mathrm{V}} \mathbb{I})^{-1/2} \left(2\Sigma_x - \Upsilon_x^{(2)} + \beta_{\mathrm{V}} \mathbb{I}\right) (3\Sigma_x + 3\beta_{\mathrm{V}} \mathbb{I})^{-1/2}}_{=:L} \preceq \mathbb{I}.$$

We deduce that all eigenvalues of the matrix L lie in the interval $[0, 1]$ and hence all the eigenvalues of the matrix L^2 belong to the interval $[0, 1]$. As a result, we have

$$\left(2\Sigma_x - \Upsilon_x^{(2)} + \beta_{\mathrm{V}} \mathbb{I}\right) (3\Sigma_x + 3\beta_{\mathrm{V}} \mathbb{I})^{-1} \left(2\Sigma_x - \Upsilon_x^{(2)} + \beta_{\mathrm{V}} \mathbb{I}\right) \preceq (3\Sigma_x + 3\beta_{\mathrm{V}} \mathbb{I}).$$

Thus, we obtain

$$\theta_x^\top \left(2\Sigma_x - \Upsilon_x^{(2)} + \beta_{\mathrm{V}} \mathbb{I}\right) (\Sigma_x + \beta_{\mathrm{V}} \mathbb{I})^{-1} \left(2\Sigma_x - \Upsilon_x^{(2)} + \beta_{\mathrm{V}} \mathbb{I}\right) \theta_x \leq 9\theta_x^\top (\Sigma_x + \beta_{\mathrm{V}} \mathbb{I}) \theta_x. \tag{B.19}$$

Finally, applying Lemma 23 and combining bounds (B.18) and (B.19) yields the claim.

B.3.3 Proof of claim (4.29b)

The quantity of interest can be written as

$$\|z-x\|_z^2 - \|z-x\|_x^2 = \sum_{i=1}^{n} \left(a_i^\top (z-x)\right)^2 (\varphi_{z,i} - \varphi_{x,i}) .$$

We can write $z = x + \alpha u$, where α is a scalar and u is a unit vector in $\mathbb{R}^d$. Then we have

$$\|z-x\|_z^2 - \|z-x\|_x^2 = \alpha^2 \sum_{i=1}^{n} \left(a_i^\top u\right)^2 (\varphi_{z,i} - \varphi_{x,i}) .$$

We apply a Taylor series expansion for $\sum_{i=1}^n \left(a_i^\top u\right)^2 (\varphi_{z,i} - \varphi_{x,i})$ around the point x, along the line u. There exists a point $y \in \overline{xz}$ such that

$$\sum_{i=1}^{n} \left(a_i^\top u\right)^2 (\varphi_{z,i} - \varphi_{x,i}) = \sum_{i=1}^{n} \left(a_i^\top u\right)^2 \left((z-x)^\top \nabla \varphi_{x,i} + \frac{1}{2} (z-x)^\top \nabla^2 \varphi_{y,i} (z-x) \right) .$$

Multiplying both sides by α^2, and using the shorthand $\eta_{x,i} = \frac{a_i^\top (z-x)}{s_{x,i}}$, we obtain

$$\|z-x\|_z^2 - \|z-x\|_x^2 = \sum_{i=1}^{n} \eta_{x,i}^2 s_{x,i}^2 (z-x)^\top \nabla \varphi_{x,i} + \sum_{i=1}^{n} \eta_{x,i}^2 s_{x,i}^2 \frac{1}{2} (z-x)^\top \nabla^2 \varphi_{y,i} (z-x) . \tag{B.20}$$

Substituting the expression for $\nabla \varphi_{x,i}$ from Lemma 25(b) in equation (B.20) and performing some algebra, the first term on the RHS of equation (B.20) can be written as

$$\sum_{i=1}^{n} \eta_{x,i}^2 s_{x,i}^2 (z-x)^\top \nabla \varphi_{x,i} = 2 \sum_{i=1}^{n} \left(\frac{7}{3} \sigma_{x,i} + \beta_{\mathrm{V}} \right) \eta_{x,i}^3 - \frac{1}{3} \sum_{i,j=1}^{n} \sigma_{x,i,j}^2 (\eta_{x,i} + \eta_{x,j})^3 . \tag{B.21}$$

On the other hand, using Lemma 25 (d), we have

$$\frac{1}{2} s_{x,i}^2 \left| (z-x)^\top \nabla^2 \varphi_{y,i} (z-x) \right| \leq \frac{s_{x,i}^2}{s_{y,i}^2} \left[14 (\sigma_{y,i} + \beta_{\mathrm{V}}) \frac{s_{x,i}^2}{s_{y,i}^2} \eta_{x,i}^2 + 11 \left(\sum_{j=1}^{n} \sigma_{y,i,j}^2 \eta_{x,j}^2 \frac{s_{x,j}^2}{s_{y,j}^2} \right) \right] . \tag{B.22}$$

Now, we use a fourth degree Gaussian polynomial to bound both the terms on the RHS of inequality (B.22). To do so, we use high probability bound for $s_{x,i}/s_{y,i}$. In particular, we use the high probability bounds for the events $\mathcal{E}_1$ and $\mathcal{E}_2$ defined in equations (B.13a)

and (B.13b). Multiplying both sides of inequality (B.22) by $\eta_{x,i}^2$ and summing over the index i, we obtain that with probability at least $1-\epsilon/4$, we have

$$\begin{aligned}
&\sum_{i=1}^{n} \eta_{x,i}^2 s_{x,i}^2 \left| \frac{1}{2} (z-x)^\top \nabla^2 \varphi_{y,i} (z-x) \right| \\
&\quad \leq \quad \left[14 \sum_{i=1}^{n} (\sigma_{y,i} + \beta_{\mathrm{V}}) \frac{s_{x,i}^4}{s_{y,i}^4} \eta_{x,i}^4 + 11 \sum_{i,j=1}^{n} \sigma_{y,i,j}^2 \eta_{x,i}^2 \eta_{x,j}^2 \frac{s_{x,i}^2 s_{x,j}^2}{s_{y,i}^2 s_{y,j}^2} \right] \\
&\overset{\text{(hpb.(B.13a))}}{\leq} \quad \tau^4 \left[14 \sum_{i=1}^{n} (\sigma_{y,i} + \beta_{\mathrm{V}}) \eta_{x,i}^4 + 11 \sum_{i,j=1}^{n} \sigma_{y,i,j}^2 \eta_{x,i}^2 \eta_{x,j}^2 \right] \\
&\overset{\text{(AM-GM)}}{\leq} \quad \tau^4 \left[14 \sum_{i=1}^{n} (\sigma_{y,i} + \beta_{\mathrm{V}}) \eta_{x,i}^4 + \frac{11}{2} \sum_{i,j=1}^{n} \sigma_{y,i,j}^2 (\eta_{x,i}^4 + \eta_{x,j}^4) \right] \\
&\overset{\text{(Lem. 23(a))}}{\leq} \quad 25 \tau^4 \sum_{i=1}^{n} (\sigma_{y,i} + \beta_{\mathrm{V}}) \eta_{x,i}^4 \\
&\overset{\text{(hpb.(B.13b))}}{\leq} \quad 50 \sum_{i=1}^{n} (\sigma_{x,i} + \beta_{\mathrm{V}}) \eta_{x,i}^4,
\end{aligned} \tag{B.23}$$

where "hpb" stands for high probability bound for events $\mathcal{E}_1$ and $\mathcal{E}_2$. In the last step, we have used the fact that $\tau^6/(2-\tau)^2 \leq 2$ for $\tau = 1.05$. Combining equations (B.20), (B.21) and (B.23) and noting that $\eta_{x,i} = r\hat{a}_i^\top \xi/(nd)^{1/4}$, we find that

$$\begin{aligned}
&\left| \|z-x\|_z^2 - \|z-x\|_x^2 \right| \\
&\leq \frac{14}{3} \left| \sum_{i=1}^{n} (\sigma_{x,i} + \beta_{\mathrm{V}}) \eta_{x,i}^3 \right| + \frac{8}{3} \left| \sum_{i,j=1}^{n} \sigma_{x,i,j}^2 \left((\eta_{x,i} + \eta_{x,j})/2 \right)^3 \right| + 38 \sum_{i=1}^{n} \sigma_{x,i} \eta_{x,i}^4 \\
&\leq \frac{14}{3} \frac{r^3}{(nd)^{3/4}} \left| \sum_{i=1}^{n} (\sigma_{x,i} + \beta_{\mathrm{V}}) \left(\hat{a}_{x,i}^\top \xi \right)^3 \right| + \frac{8}{3} \frac{r^3}{(nd)^{3/4}} \left| \sum_{i,j=1}^{n} \sigma_{x,i,j}^2 \left(\frac{1}{2} (\hat{a}_{x,i} + \hat{a}_{x,j})^\top \xi \right)^3 \right| \\
&\quad + 50 \frac{r^4}{nd} \sum_{i=1}^{n} (\sigma_{x,i} + \beta_{\mathrm{V}}) (\hat{a}_{x,i}^\top \xi)^4,
\end{aligned} \tag{B.24}$$

where the last step follows from the fact that $0 \leq \sigma_{x,i} \leq \sigma_{x,i} + \beta_{\mathrm{V}}$. In order to show that $\left| \|z-x\|_z^2 - \|z-x\|_x^2 \right|$ is bounded as $O\left(1/\sqrt{nd}\right)$ with high probability, it suffices to show that with high probability, the third and fourth degree polynomials of $\hat{a}_{x,i}^\top \xi$, that appear in bound (B.24), are bounded by $O\left((nd)^{1/4}\right)$ and $O\left(\sqrt{nd}\right)$ respectively.

Applying the bounds (B.8b), (B.8c) and (B.8d) from Lemma 27, we have with probability at least $1-\epsilon$,

$$\|z-x\|_z^2 - \|z-x\|_x^2 \leq \frac{r^3}{\sqrt{nd}} \left(\frac{22\sqrt{15}\chi_3}{3} \right) + \frac{r^4}{\sqrt{nd}} \left(50\sqrt{105}\chi_4 \right).$$

Using the condition

$$r \leq \min\left\{\frac{\epsilon}{22\sqrt{5/3}\chi_3}, \sqrt{\frac{\epsilon}{50\sqrt{105}\chi_4}}\right\}, \tag{B.25}$$

completes our proof of claim (4.29b).

B.3.4 Proof of Lemma 26

The proof is based on Lemma 24 and a simple application of the standard chi-square tail bounds. According to Lemma 24, we have that for $v \in \overline{xz}$,

$$\left|1 - \frac{s_{v,i}}{s_{x,i}}\right| \leq \left(\frac{n}{d}\right)^{\frac{1}{4}} \|x - v\|_x \leq \left(\frac{n}{d}\right)^{\frac{1}{4}} \|x - z\|_x .$$

According to equation (B.9), the proposal follows Gaussian distribution

$$\left(\frac{n}{d}\right)^{\frac{1}{4}} \|x - z\|_x = \frac{r}{d^{1/2}} \|\xi\|_2 ,$$

where $\xi \sim \mathcal{N}(0, \mathbb{I}_d)$. Using the standard chi-square tail bound we have that for $\delta > 0$,

$$\mathbb{P}\left[\|\xi\|_2 / \sqrt{d} \geq 1 + \delta\right] \leq \exp\left(-d\delta^2/2\right).$$

Plugging in $\delta = \sqrt{\frac{2}{d}} \log^{\frac{1}{2}}\left(\frac{4}{\epsilon}\right)$ concludes the lemma.

B.3.5 Proof of Lemma 27

The proof relies on the classical fact that the tails of a polynomial in Gaussian random variables decay exponentially independently of dimension. In particular, Theorem 6.7 by [87] ensures that for any integers $d, k \geq 1$, any polynomial $f : \mathbb{R}^d \to \mathbb{R}$ of degree k, and any scalar $t \geq (2e)^{k/2}$, we have

$$\mathbb{P}\left[|f(\xi)| \geq t\left(\mathbb{E}f(\xi)^2\right)^{\frac{1}{2}}\right] \leq \exp\left(-\frac{k}{2e}t^{2/k}\right), \tag{B.26}$$

where $\xi \sim \mathcal{N}(0, \mathbb{I}_n)$ denotes a standard Gaussian vector in n dimensions. Also, the following observations on the behavior of the vectors $\hat{a}_{x,i}$ defined in equation (B.7) are useful:

$$\|\hat{a}_{x,i}\|_2^2 = \theta_{x,i} \overset{(i)}{\leq} \sqrt{\frac{n}{d}} \quad \text{for all } i \in [n], \quad \text{and} \tag{B.27a}$$

$$(\hat{a}_{x,i}^\top \hat{a}_{x,j})^2 = \theta_{x,i,j}^2 \quad \text{for all } i, j \in [n], \tag{B.27b}$$

where inequality (i) follows from Lemma 11 (c).

Proof of bound (B.8a)

We have

$$\begin{aligned}
&\mathbb{E}\left(\sum_{i=1}^{n}\left(\sigma_{x,i}+\beta_{\mathrm{V}}\right)\left(\hat{a}_{x,i}^{\top}\xi\right)^{2}\right)^{2} \\
&=\sum_{i,j=1}^{n}\left(\sigma_{x,i}+\beta_{\mathrm{V}}\right)\left(\sigma_{x,j}+\beta_{\mathrm{V}}\right)\mathbb{E}\left(\hat{a}_{x,i}^{\top}\xi\right)^{2}\left(\hat{a}_{x,j}^{\top}\xi\right)^{2} \\
&=\sum_{i,j=1}^{n}\left(\sigma_{x,i}+\beta_{\mathrm{V}}\right)\left(\sigma_{x,j}+\beta_{\mathrm{V}}\right)\left(\|\hat{a}_{x,i}\|_{2}^{2}\|\hat{a}_{x,j}\|_{2}^{2}+2\left(\hat{a}_{x,i}^{\top}\hat{a}_{x,j}\right)^{2}\right) \\
&=\sum_{i,j=1}^{n}\left(\sigma_{x,i}+\beta_{\mathrm{V}}\right)\left(\sigma_{x,j}+\beta_{\mathrm{V}}\right)\left(\theta_{x,i}\theta_{x,j}+2\theta_{x,i,j}^{2}\right) \\
&\overset{(i)}{=} d^{2}+2d \\
&\leq 3d^{2},
\end{aligned}$$

where step (i) follows from properties (c) and (d) from Lemma 23. Applying the bound (B.26) with $k=2, t=e\log(\frac{4}{\epsilon})$ yields the claim. We verify that for $\epsilon\in(0,1/15]$, $t\geq 2e$.

Proof of bound (B.8b)

Using Isserlis' theorem [86] for Gaussian moments, we obtain

$$\begin{aligned}
\mathbb{E}\left(\sum_{i=1}^{n}\left(\sigma_{x,i}+\beta_{\mathrm{V}}\right)\left(\hat{a}_{x,i}^{\top}\xi\right)^{3}\right)^{2} &= \sum_{i,j=1}^{n}\left(\sigma_{x,i}+\beta_{\mathrm{V}}\right)\left(\sigma_{x,i}+\beta_{\mathrm{V}}\right)\mathbb{E}\left(\hat{a}_{x,i}^{\top}\xi\right)^{3}\left(\hat{a}_{x,j}^{\top}\xi\right)^{3} \\
&= 9\underbrace{\sum_{i,j=1}^{n}\left(\sigma_{x,i}+\beta_{\mathrm{V}}\right)\left(\sigma_{x,j}+\beta_{\mathrm{V}}\right)\|\hat{a}_{x,i}\|_{2}^{2}\|\hat{a}_{x,j}\|_{2}^{2}\left(\hat{a}_{x,i}^{\top}\hat{a}_{x,j}\right)}_{=:N_1} \\
&\quad + 6\underbrace{\sum_{i,j=1}^{n}\left(\sigma_{x,i}+\beta_{\mathrm{V}}\right)\left(\sigma_{x,j}+\beta_{\mathrm{V}}\right)\left(\hat{a}_{x,i}^{\top}\hat{a}_{x,j}\right)^{3}}_{=:N_2}. \qquad \text{(B.28)}
\end{aligned}$$

We claim that the two terms in this sum are bounded as $N_1\leq\sqrt{nd}$ and $N_2\leq\sqrt{nd}$. Assuming the claims as given, we now complete the proof. Plugging in the bounds for N_1 and N_2 in equation (B.28) we find that $\mathbb{E}\big(\sum_{i=1}^{n}\left(\sigma_{x,i}+\beta_{\mathrm{V}}\right)\left(\hat{a}_{x,i}^{\top}\xi\right)^{3}\big)^{2}\leq 15\sqrt{nd}$. Applying the bound (B.26) with $k=3, t=\left(\frac{2e}{3}\log(4/\epsilon)\right)^{3/2}$ yields the claim. We also verify that for $\epsilon\in(0,1/15]$, $t\geq(2e)^{3/2}$. We now turn to proving the bounds on N_1 and N_2.

Bounding N_1: Let B be an $n \times d$ matrix with its i-th row given by $\sqrt{(\sigma_{x,i} + \beta_{\mathrm{V}})}\hat{a}_{x,i}^\top$. Observe that

$$\sum_{i=1}^{n} (\sigma_{x,i} + \beta_{\mathrm{V}})\, \hat{a}_i \hat{a}_{x,i}^\top = V_x^{-1/2} \left(\sum_{i=1}^{n} (\sigma_{x,i} + \beta_{\mathrm{V}})\, \frac{a_i a_i^\top}{s_{x,i}^2} \right) V_x^{-1/2} = V_x^{-1/2} V_x V_x^{-1/2} = \mathbb{I}_d. \tag{B.29}$$

Thus we have $B^\top B = \mathbb{I}_d$, which implies that $BB^\top$ is an orthogonal projection matrix. Letting $v \in \mathbb{R}^n$ be a vector such that $v_i = \sqrt{(\sigma_{x,i} + \beta_{\mathrm{V}})}\, \|\hat{a}_{x,i}\|_2^2$, we then have

$$\begin{aligned}
&\sum_{i,j=1}^{n} (\sigma_{x,i}+\beta_{\mathrm{V}}) \|\hat{a}_{x,i}\|_2^2\, \hat{a}_{x,i}^\top\, (\sigma_{x,j}+\beta_{\mathrm{V}}) \|\hat{a}_{x,j}\|_2^2\, \hat{a}_{x,j} \\
&= \left\| \sum_{i=1}^{n} (\sigma_{x,i}+\beta_{\mathrm{V}}) \|\hat{a}_{x,i}\|_2^2\, \hat{a}_{x,i} \right\|_2^2 \\
&= \left\| B^\top v \right\|_2^2 \\
&\overset{(i)}{\leq} \|v\|_2^2 ,
\end{aligned}$$

where inequality (i) follows from the fact that $v^\top P v \leq \|v\|_2^2$ for any orthogonal projection matrix P. Equation (B.27a) implies that $v_i^2 = (\sigma_{x,i} + \beta_{\mathrm{V}})\, \theta_{x,i}^2$. Using Lemma 23(e), we find that

$$\|v\|_2^2 = \sum_{i=1}^{n} (\sigma_{x,i} + \beta_{\mathrm{V}})\, \theta_{x,i}^2 \leq \sqrt{nd}.$$

Bounding N_2: We see that

$$\begin{aligned}
&\sum_{i,j=1}^{n} (\sigma_{x,i} + \beta_{\mathrm{V}})\, (\sigma_{x,j} + \beta_{\mathrm{V}})\, \left(\hat{a}_{x,i}^\top \hat{a}_{x,j}\right)^3 \\
&\overset{\text{(C–S)}}{\leq} \sum_{i,j=1}^{n} (\sigma_{x,i} + \beta_{\mathrm{V}})\, (\sigma_{x,j} + \beta_{\mathrm{V}})\, \left(\hat{a}_{x,i}^\top \hat{a}_{x,j}\right)^2 \|\hat{a}_{x,i}\|_2\, \|\hat{a}_{x,j}\|_2 \\
&\overset{\text{(eqns.(B.27a),(B.27b))}}{\leq} \sum_{i,j=1}^{n} (\sigma_{x,i} + \beta_{\mathrm{V}})\, (\sigma_{x,j} + \beta_{\mathrm{V}})\, \theta_{x,i,j}^2 \sqrt{\theta_{x,i}\theta_{x,j}} \\
&\overset{\text{(Lem. 11(c))}}{\leq} \sqrt{\frac{n}{d}} \sum_{i,j=1}^{n} (\sigma_{x,i} + \beta_{\mathrm{V}})\, (\sigma_{x,j} + \beta_{\mathrm{V}})\, \theta_{x,i,j}^2.
\end{aligned}$$

We now apply Lemma 23(d) followed by Lemma 23(c) to obtain the claimed bound on N_2.

Proof of bound (B.8c)

Let $c_{i,j} = \dfrac{(\hat{a}_{x,i} + \hat{a}_{x,j})}{2}$ for $i, j \in [n]$. Using Isserlis' theorem for Gaussian moments, we obtain

$$\begin{aligned}
&\mathbb{E}\left(\sum_{i,j=1}^{n} \sigma_{x,i,j}^2 \left(c_{i,j}^\top \xi\right)^3\right)^2 \\
&= \sum_{i,j,k,l=1}^{n} \sigma_{x,i,j}^2 \sigma_{x,k,l}^2 \mathbb{E}\left(c_{i,j}^\top \xi\right)^3 \left(c_{k,l}^\top \xi\right)^3 \\
&= 9 \underbrace{\sum_{i,j,k,l=1}^{n} \sigma_{x,i,j}^2 \sigma_{x,k,l}^2 \|c_{i,j}\|_2^2 \|c_{k,l}\|_2^2 \left(c_{i,j}^\top c_{k,l}\right)}_{=: C_1} + 6 \underbrace{\sum_{i,j,k,l=1}^{n} \sigma_{x,i,j}^2 \sigma_{x,k,l}^2 \left(c_{i,j}^\top c_{k,l}\right)^3}_{=: C_2}
\end{aligned}$$

We claim that $C_1 \leq \sqrt{nd}$ and $C_2 \leq \sqrt{nd}$. Assuming the claims as given, the result follows using similar arguments as in the previous part. We now bound $C_i, i = 1, 2$, using arguments similar to the ones used in Section B.3.5 to bound $N_i, i = 1, 2$, respectively. The following bounds on $\|c_{i,j}\|_2^2$ are used in the arguments that follow:

$$\|c_{i,j}\|_2^2 \overset{\text{SSI}}{\leq} \frac{1}{2}\left(\|\hat{a}_i\|_2^2 + \|\hat{a}_j\|_2^2\right) = \frac{1}{2}\left(\theta_{x,i} + \theta_{x,j}\right) \tag{B.30a}$$

$$\overset{\text{Lem. 11(c)}}{\leq} \sqrt{\frac{n}{d}}. \tag{B.30b}$$

Bounding C_1: Let B be the same $n \times d$ matrix as in the proof of previous part with its i-th row given by $\sqrt{(\sigma_{x,i} + \beta_{\mathrm{V}})}\hat{a}_{x,i}^\top$. Define the vector $u \in \mathbb{R}^d$ with entries given by $u_i = \sum_{j=1}^n \sigma_{x,i,j}^2 \|c_{i,j}\|_2^2 / (\sigma_{x,i} + \beta_{\mathrm{V}})^{1/2}$. We have

$$\begin{aligned}
&\sum_{i,j,k,l=1}^{n} \sigma_{x,i,j}^2 \sigma_{x,k,l}^2 \|c_{i,j}\|_2^2 \|c_{k,l}\|_2^2 \left(c_{i,j}^\top c_{k,l}\right) \\
&\leq \left\| \sum_{i,j=1}^{n} \sigma_{x,i,j}^2 \|c_{i,j}\|_2^2 c_{i,j} \right\|_2^2 \\
&\overset{\text{(SSI)}}{\leq} \frac{1}{2}\left(\left\| \sum_{i,j=1}^{n} \sigma_{x,i,j}^2 \|c_{i,j}\|_2^2 \hat{a}_{x,i} \right\|_2^2 + \left\| \sum_{i,j=1}^{n} \sigma_{x,i,j}^2 \|c_{i,j}\|_2^2 \hat{a}_{x,j} \right\|_2^2 \right) \\
&= \left\| B^\top u \right\|_2^2 \\
&\overset{(i)}{\leq} \|u\|_2^2,
\end{aligned}$$

where inequality (i) follows from the fact that $v^\top P v \leq \|v\|_2^2$ for any orthogonal projection matrix P. It is left to bound the term u_i^2. We see that

$$\begin{aligned} u_i^2 &= \frac{1}{\sigma_{x,i} + \beta_{\mathrm{V}}} \sum_{j,k=1}^{n} \sigma_{x,i,j}^2 \sigma_{x,i,k}^2 \left\|c_{i,j}\right\|_2^2 \left\|c_{i,k}\right\|_2^2 \\ &\overset{\text{(bnd. (B.30b))}}{\leq} \sqrt{\frac{n}{d}} \frac{1}{\sigma_{x,i} + \beta_{\mathrm{V}}} \sum_{j,k=1}^{n} \sigma_{x,i,j}^2 \sigma_{x,i,k}^2 \left\|c_{i,j}\right\|_2^2 \\ &\overset{\text{(Lem. 23(a))}}{\leq} \sqrt{\frac{n}{d}} \frac{\sigma_{x,i}}{\sigma_{x,i} + \beta_{\mathrm{V}}} \sum_{j=1}^{n} \sigma_{x,i,j}^2 \left\|c_{i,j}\right\|_2^2 \\ &\overset{\text{(bnd. (B.30a))}}{\leq} \sqrt{\frac{n}{d}} \sum_{j=1}^{n} \sigma_{x,i,j}^2 \frac{\theta_{x,i} + \theta_{x,j}}{2}. \end{aligned}$$

Now, summing over i and using symmetry of indices i, j, we find that

$$\|u\|_2^2 \leq \sqrt{\frac{n}{d}} \sum_{i=1}^{n} \sum_{j=1}^{n} \sigma_{x,i,j}^2 \theta_{x,i} \overset{\text{(Lem. 23(a))}}{=} \sqrt{\frac{n}{d}} \sum_{i=1}^{n} \sigma_{x,i} \theta_{x,i} \overset{\text{(Lem. 23(c))}}{\leq} \sqrt{nd},$$

thereby implying that $C_1 \leq \sqrt{nd}$.

Bounding C_2: Using the Cauchy-Schwarz inequality and the bound (B.30b), we find that

$$\begin{aligned} \sum_{i,j,k,l=1}^{n} \sigma_{x,i,j}^2 \sigma_{x,k,l}^2 \left(c_{i,j}^\top c_{k,l}\right)^3 &\leq \sum_{i,j,k,l=1}^{n} \sigma_{x,i,j}^2 \sigma_{x,k,l}^2 \left(c_{i,j}^\top c_{k,l}\right)^2 \left\|c_{i,j}\right\|_2 \left\|c_{k,l}\right\|_2 \\ &\leq \sqrt{\frac{n}{d}} \sum_{i,j,k,l=1}^{n} \sigma_{x,i,j}^2 \sigma_{x,k,l}^2 \left(c_{i,j}^\top c_{k,l}\right)^2. \end{aligned}$$

Using SSI and the symmetry of pairs of indices (i, j) and (k, l), we obtain

$$\sum_{i,j,k,l=1}^{n} \sigma_{x,i,j}^2 \sigma_{x,k,l}^2 \left(c_{i,j}^\top c_{k,l}\right)^2 \leq \sum_{i,j,k,l=1}^{n} \sigma_{x,i,j}^2 \sigma_{x,k,l}^2 \left(\hat{a}_{x,i}^\top \hat{a}_k\right)^2 = \sum_{i,k=1}^{n} \sigma_{x,i} \sigma_{x,k} \left(\hat{a}_{x,i}^\top \hat{a}_k\right)^2.$$

The resulting expression can be bounded as follows:

$$\sum_{i,k=1}^{n} \sigma_{x,i} \sigma_{x,k} \left(\hat{a}_{x,i}^\top \hat{a}_k\right)^2 \overset{\text{(eqn. (B.27b))}}{=} \sum_{i,k=1}^{n} \sigma_{x,i} \sigma_{x,k} \theta_{x,i,k}^2 \overset{\text{(Lem. 23(d))}}{\leq} \sum_{i=1}^{n} \sigma_{x,i} \theta_{x,i} \overset{\text{(Lem. 23(c))}}{\leq} n.$$

Putting the pieces together yields the claimed bound on C_2.

Proof of bound (B.8d)

Observe that $\hat{a}_{x,i}^\top \xi \sim \mathcal{N}(0, \theta_{x,i})$ and hence $\mathbb{E}\left(\hat{a}_{x,i}^\top \xi\right)^8 = 105\,\theta_{x,i}^4$. Thus we have

$$\begin{aligned}
\mathbb{E}\left(\sum_{i=1}^n \sigma_{x,i}\left(\hat{a}_{x,i}^\top \xi\right)^4\right)^2 &\overset{\text{C–S}}{\leq} \sum_{i,j=1}^n \sigma_{x,i}\sigma_{x,j}\left(\mathbb{E}\left(\hat{a}_{x,i}^\top \xi\right)^8\right)^{\frac{1}{2}}\left(\mathbb{E}\left(\hat{a}_{x,j}^\top \xi\right)^8\right)^{\frac{1}{2}} \\
&= 105\sum_{i,j=1}^n \sigma_{x,i}\sigma_{x,j}\theta_{x,i}^2\theta_{x,j}^2 \\
&= 105\left(\sum_{i=1}^n \sigma_{x,i}\theta_{x,i}^2\right)^2 \\
&\overset{\text{(Lem. 23}(e)\text{)}}{\leq} 105nd.
\end{aligned}$$

Applying the bound (B.26) with $k = 4, t = \left(\frac{e}{2}\log(4/\epsilon)\right)^2$ yields the result. We also verify that for $\epsilon \in (0, 1/15]$, we have $t \geq (2e)^2$

B.3.6 Proof of Lemma 25

We now derive the different expressions for derivatives and prove the bounds for Hessians of $x \mapsto \varphi_{x,i}$, $i \in [n]$ and $x \mapsto \Psi_x$. In this section we use the simpler notation $H_x := \nabla^2 \mathcal{F}_x$.

Gradient of σ

Using $s_{x+h,i} = (b_i - a_i^\top(x+h)) = s_{x,i} - a_i^\top h$, we define the Hessian difference matrix

$$\Delta_{x,h}^H := H_{x+h} - H_x = \sum_{i=1}^n a_i a_i^\top \left(\frac{1}{(s_{x,i} - a_i^\top h)^2} - \frac{1}{s_{x,i}^2}\right). \tag{B.31}$$

Up to second order terms, we have

$$\frac{1}{s_{x+h,i}^2} = \frac{1}{s_{x,i}^2}\left[1 + \frac{2a_i^\top h}{s_{x,i}} + \frac{3(a_i^\top h)^2}{s_{x,i}^2}\right] + O\left(\|h\|_2^3\right), \tag{B.32a}$$

$$\Delta_{x,h}^H = \sum_{i=1}^n \frac{a_i a_i^\top}{s_{x,i}^2}\left[\frac{2a_i^\top h}{s_{x,i}} + \frac{3(a_i^\top h)^2}{s_{x,i}^2}\right] + O\left(\|h\|_2^3\right), \tag{B.32b}$$

$$a_i^T H_{x+h}^{-1} a_i = a_i^\top H_x^{-1} a_i - a_i^\top H_x^{-1}\Delta_{x,h}^H H_x^{-1} a_i + a_i^\top H_x^{-1}\Delta_{x,h}^H H_x^{-1}\Delta_{x,h}^H H_x^{-1} a_i + O\left(\|h\|_2^3\right). \tag{B.32c}$$

Collecting different first order terms in $\sigma_{x+h,i} - \sigma_{x,i}$, we obtain

$$\begin{aligned}\sigma_{x+h,i} - \sigma_{x,i} &= 2\,\frac{a_i^\top H_x^{-1} a_i}{s_{x,i}^2}\frac{a_i^\top h}{s_{x,i}} - 2\,\frac{a_i^\top H_x^{-1}\left(\sum_{j=1}^n \frac{a_j a_j^\top}{s_{x,j}^2}\frac{a_j^\top h}{s_{x,j}}\right) H_x^{-1} a_i}{s_{x,i}^2} + O\left(\|h\|_2^2\right)\\ &= 2\left[\sigma_{x,i}\frac{a_i^\top h}{s_{x,i}} - \sum_{j=1}^n \sigma_{x,i,j}^2\,\frac{a_j^\top h}{s_{x,j}}\right] + O\left(\|h\|_2^2\right)\\ &= 2\left[(\Sigma_x - \Upsilon_x^{(2)})S_x^{-1}A\right]_i h + O\left(\|h\|_2^2\right).\end{aligned}$$

Dividing both sides by h and letting $h \to 0$ yields the claim.

Gradient of φ

Using the chain rule and the fact that $\nabla s_{x,i} = -a_i$, we find that

$$\begin{aligned}\nabla\varphi_{x,i} &= \frac{\nabla\sigma_{x,i}}{s_{x,i}^2} - 2\,(\sigma_{x,i} + \beta_{\mathrm{V}})\frac{\nabla s_{x,i}}{s_{x,i}^3}\\ &= \frac{2}{s_{x,i}^2}A^\top S_x^{-1}\left[2\Sigma_x + \beta_{\mathrm{V}}\,\mathbb{I} - \Upsilon_x^{(2)}\right]e_i,\end{aligned}$$

as claimed.

Gradient of Ψ

For convenience, let us restate equations (B.7) and (B.29):

$$\hat{a}_{x,i} = \frac{1}{s_{x,i}}V_x^{-1/2}a_i, \quad \text{and} \quad \sum_{i=1}^n (\sigma_{x,i} + \beta_{\mathrm{V}})\,\hat{a}_{x,i}\hat{a}_{x,i}^\top = \mathbb{I}_d.$$

For a unit vector h, we have

$$\begin{aligned}&h^\top \nabla \log\det V_x\\ &= \lim_{\delta\to 0}\frac{1}{\delta}\left[\operatorname{trace}\log\left(\sum_{i=1}^n \frac{(\sigma_{x+\delta h,i} + \beta_{\mathrm{V}})}{\left(1 - \delta a_i^\top h/s_{x,i}\right)^2}\hat{a}_{x,i}\hat{a}_{x,i}^\top\right) - \operatorname{trace}\log\left(\sum_{i=1}^n (\sigma_{x,i} + \beta_{\mathrm{V}})\,\hat{a}_{x,i}\hat{a}_{x,i}^\top\right)\right].\end{aligned} \tag{B.33}$$

Let $\log L$ denote the logarithm of the matrix L. Keeping track of the first order terms on RHS of equation (B.33), we find that

$$\begin{aligned}
&\text{trace}\left[\log\left(\sum_{i=1}^{n}\left(\sigma_{x+\delta h,i}+\beta_{\mathrm{V}}\right)\frac{\hat{a}_{x,i}\hat{a}_{x,i}^{\top}}{\left(1-\delta a_i^{\top}h/s_{x,i}\right)^2}\right)\right]-\text{trace}\left[\log\left(\sum_{i=1}^{n}\left(\sigma_{x,i}+\beta_{\mathrm{V}}\right)\hat{a}_{x,i}\hat{a}_{x,i}^{\top}\right)\right]\\
&=\text{trace}\left[\log\left(\sum_{i=1}^{n}\left(\sigma_{x+\delta h,i}+\beta_{\mathrm{V}}+\delta h^{\top}\nabla\sigma_{x,i}\right)\left(1+2\delta\frac{a_i^{\top}h}{s_{x,i}^2}\right)\right)\right]\\
&-\text{trace}\left[\log\left(\sum_{i=1}^{n}\left(\sigma_{x,i}+\beta_{\mathrm{V}}\right)\hat{a}_{x,i}\hat{a}_{x,i}^{\top}\right)\right]+O\left(\delta^2\right)\\
&=\text{trace}\left[\sum_{i=1}^{n}\delta\left(2\left(\sigma_{x,i}+\beta_{\mathrm{V}}\right)\frac{a_i^{\top}h}{s_{x,i}^2}+h^{\top}\nabla\sigma_{x,i}\right)\hat{a}_{x,i}\hat{a}_{x,i}^{\top}\right]+O\left(\delta^2\right)\\
&=\delta\left(\sum_{i=1}^{n}\left(2\left(\sigma_{x,i}+\beta_{\mathrm{V}}\right)\frac{a_i^{\top}h}{s_{x,i}^2}+h^{\top}\nabla\sigma_{x,i}\right)\theta_i\right)+O\left(\delta^2\right),
\end{aligned}$$

where we have used the fact $\text{trace}(\log\mathbb{I})=0$. Letting $\delta\to 0$ and substituting expression of $h^{\top}\nabla\sigma_x$ from part (a), we obtain

$$h^{\top}\nabla\log\det V_x=A_x^{\top}\left(4\Sigma_x+2\beta_{\mathrm{V}}\mathbb{I}-2\Upsilon_x^{(2)}\right)\Theta_x h.$$

Bound on Hessian $\nabla^2\varphi$

In terms of the shorthand $E_{ii}=e_ie_i^{\top}$, we claim that for any $h\in\mathbb{R}^d$,

$$\begin{aligned}
h^{\top}\nabla^2\varphi_{x,i}h=\frac{2}{s_{x,i}^2}h^{\top}A_x^{\top}\Big[&E_{ii}\left(3\left(\Sigma_x+\beta_{\mathrm{V}}\mathbb{I}\right)+7\Sigma_x-8\,\text{diag}(\Upsilon_x^{(2)}e_i)\right)E_{ii}\\
&+\text{diag}(\Upsilon_x e_i)(4\Upsilon_x-3\mathbb{I})\,\text{diag}(\Upsilon_x e_i)\Big]A_x h.
\end{aligned}\tag{B.34}$$

Note that

$$\varphi_{x+h,i}-\varphi_{x,i}=\underbrace{\left(\frac{a_i^{\top}H_{x+h,i}^{-1}a_i}{s_{x+h,i}^4}-\frac{a_i^{\top}H_{x,i}^{-1}a_i}{s_{x,i}^4}\right)}_{=:A_1}+\underbrace{\beta_{\mathrm{V}}\left(\frac{1}{s_{x+h,i}^2}-\frac{1}{s_{x,i}^2}\right)}_{=:A_2}.\tag{B.35}$$

The second order Taylor expansion of $1/s_{x,i}^4$ is given by

$$\frac{1}{s_{x+h,i}^4}=\frac{1}{s_{x,i}^4}\left[1+\frac{4a_i^{\top}h}{s_{x,i}}+\frac{10(a_i^{\top}h)^2}{s_{x,i}^2}\right]+O\left(\|h\|_2^3\right).$$

Let B_1 and B_2 denote the second order terms, i.e., the terms that are of order $O\left(\|h\|_2^2\right)$, in Taylor expansion of A_1 and A_2 around x, respectively. Borrowing terms from equations (B.32a)-(B.32c) and simplifying we obtain

$$\begin{aligned} B_1 &= 10\sigma_{x,i}\frac{(a_i^\top h)^2}{s_{x,i}^2} - 8\frac{a_i^\top h}{s_{x,i}}\sum_{j=1}^{n}\frac{\sigma_{x,i,j}^2}{s_{x,i}^2}\frac{a_j^\top h}{s_{x,j}} \\ &\quad -3\sum_{j=1}^{n}\frac{\sigma_{x,i,j}^2}{s_{x,i}^2}\frac{(a_j^\top h)^2}{s_{x,j}^2} + 4\sum_{j=1}^{n}\sum_{l=1}^{n}\frac{\sigma_{x,i,j}}{s_{x,i}}\sigma_{x,j,l}\frac{\sigma_{x,l,i}}{s_{x,i}}\frac{a_j^\top h}{s_{x,j}}\frac{a_l^\top h}{s_{x,l}}, \\ \text{and } B_2 &= 3\beta_{\mathrm{V}}\frac{(a_i^\top h)^2}{s_{x,i}^2}. \end{aligned}$$

Observing that the second order term in the Taylor expansion of $\varphi_{x+h,i}$ around x, is exactly $\frac{1}{2}h^\top\nabla^2\varphi_{x,i}h$ yields the claim (B.34). We now turn to prove the bound on the directional Hessian. Recall $\eta_{x,i} = a_i^\top h/s_{x,i}$. We have

$$\begin{aligned} & s_{y,i}^2\left|\frac{1}{2}h^\top\nabla^2\varphi_{x,i}h\right| \\ &= \Bigg\| 3\left(\sigma_{x,i}+\beta_{\mathrm{V}}\right)\eta_{x,i}^2 + 7\sigma_{x,i}\eta_{x,i}^2 - 8\sum_{j=1}^{n}\sigma_{x,i,j}^2\eta_{x,j}\eta_{x,i} \\ &\quad -3\sum_{j=1}^{n}\sigma_{x,i,j}^2\eta_{x,j}^2 + 4\sum_{j,k=1}^{n}\sigma_{x,i,j}\sigma_{x,j,k}\sigma_{x,k,i}\eta_{x,j}\eta_{x,k}\Bigg\| \\ &\overset{(i)}{\leq} 10\left(\sigma_{x,i}+\beta_{\mathrm{V}}\right)\eta_{x,i}^2 + 8\sum_{j=1}^{n}\sigma_{x,i,j}^2\left|\eta_{x,i}\eta_{x,j}\right| + 7\sum_{j=1}^{n}\sigma_{x,i,j}^2\eta_{x,j}^2 \\ &\overset{(ii)}{\leq} 10\left(\sigma_{x,i}+\beta_{\mathrm{V}}\right)\eta_{x,i}^2 + 4\sum_{j=1}^{n}\sigma_{x,i,j}^2\left(\eta_{x,i}^2+\eta_{x,j}^2\right) + 7\sum_{j=1}^{n}\sigma_{x,i,j}^2\eta_{x,j}^2 \\ &\overset{(iii)}{\leq} 10\left(\sigma_{x,i}+\beta_{\mathrm{V}}\right)\eta_{x,i}^2 + 4\sum_{j=1}^{n}\sigma_{x,i}\eta_{x,i}^2 + 4\sum_{j=1}^{n}\sigma_{x,i,j}^2\eta_{x,j}^2 + 7\sum_{j=1}^{n}\sigma_{x,i,j}^2\eta_{x,j}^2, \\ &\overset{(iv)}{\leq} 14\left(\sigma_{x,i}+\beta_{\mathrm{V}}\right)\eta_{x,i}^2 + 11\sum_{j=1}^{n}\sigma_{x,i,j}^2\eta_{x,j}^2, \end{aligned}$$

where step (i) follows from the fact that $\operatorname{diag}(\Upsilon_y e_i)\Upsilon_y\operatorname{diag}(\Upsilon_y e_i) \preceq \operatorname{diag}(\Upsilon_y e_i)\operatorname{diag}(\Upsilon_y e_i)$ since Υ_y is an orthogonal projection matrix; step (ii) follows from AM-GM inequality; step (iii) follows from the symmetry of indices i and j and Lemma 23(a), and step (iv) from the fact that $\sigma_{x,i} \leq \sigma_{x,i} + \beta_{\mathrm{V}}$.

Bound on Hessian $\nabla^2\Psi$

We have

$$\begin{aligned}\frac{1}{2}h^\top\left(\nabla^2\log\det V_x\right)h =&\frac{1}{2}\lim_{\delta\to 0}\frac{1}{\delta^2}\Bigg[\operatorname{trace}\log\left(\sum_{i=1}^n\frac{(\sigma_{x+\delta h,i}+\beta_{\mathrm{V}})}{\left(1-\delta a_i^\top h/s_{x,i}\right)^2}\hat{a}_{x,i}\hat{a}_{x,i}^\top\right)\\&+\operatorname{trace}\log\left(\sum_{i=1}^n\frac{(\sigma_{x-\delta h,i}+\beta_{\mathrm{V}})}{\left(1+\delta a_i^\top h/s_{x,i}\right)^2}\hat{a}_{x,i}\hat{a}_{x,i}^\top\right)\\&-2\operatorname{trace}\log\left(\sum_{i=1}^n(\sigma_x+\beta_{\mathrm{V}})\,\hat{a}_{x,i}\hat{a}_{x,i}^\top\right)\Bigg].\end{aligned}\tag{B.36}$$

Up to second order terms, we have

$$\begin{aligned}&\operatorname{trace}\left[\log\left(\sum_{i=1}^n(\sigma_{x+\delta h,i}+\beta_{\mathrm{V}})\,\frac{\hat{a}_{x,i}\hat{a}_{x,i}^\top}{\left(1-\delta a_i^\top h/s_{x,i}\right)^2}\right)\right]\\&=\operatorname{trace}\left[\log\left(\sum_{i=1}^n\left(\sigma_{x,i}+\beta_{\mathrm{V}}+\delta h^\top\nabla\sigma_{x,i}+\frac{1}{2}\delta^2h^\top\nabla^2\sigma_{x,i}h\right)\left(1+2\delta\frac{a_i^\top h}{s_{x,i}}+3\delta^2\left(\frac{a_i^\top h}{s_{x,i}}\right)^2\right)\hat{a}_{x,i}\hat{a}_{x,i}^\top\right)\right]\\&=\operatorname{trace}\left[\sum_{i=1}^n\left(\sigma_{x,i}+\beta_{\mathrm{V}}+\delta h^\top\nabla\sigma_{x,i}+\frac{1}{2}\delta^2h^\top\nabla^2\sigma_{x,i}h\right)\left(1+2\delta\frac{a_i^\top h}{s_{x,i}}+3\delta^2\left(\frac{a_i^\top h}{s_{x,i}}\right)^2\right)\hat{a}_{x,i}\hat{a}_{x,i}^\top\right]\\&\quad-\operatorname{trace}\left[\frac{1}{2}\left(\sum_{i=1}^n\left(\sigma_{x,i}+\beta_{\mathrm{V}}+\delta h^\top\nabla\sigma_{x,i}+\frac{1}{2}\delta^2h^\top\nabla^2\sigma_{x,i}h\right)\left(1+2\delta\frac{a_i^\top h}{s_{x,i}}+3\delta^2\left(\frac{a_i^\top h}{s_{x,i}}\right)^2\right)\hat{a}_{x,i}\hat{a}_{x,i}^\top\right)^2\right].\end{aligned}$$

We can similarly obtain the second order expansion of the trace of logarithmic term $\operatorname{trace}\log\left(\sum_{i=1}^n\frac{\left(\sigma_{x-\delta h,i}+\beta_{\mathrm{V}}\right)}{\left(1+\delta a_i^\top h/s_{x,i}\right)^2}\hat{a}_{x,i}\hat{a}_{x,i}^\top\right)$. Recall $\eta_{x,i}=\frac{a_i^\top h}{s_{x,i}}$. Using part (a) to substitute $h^\top\nabla\sigma_{x,i}$, we obtain

$$\begin{aligned}&\frac{1}{2}h^\top\left(\nabla^2\log\det V_x\right)h\\&=\sum_{i=1}^n\left(3\left(\sigma_{x,i}+\beta_{\mathrm{V}}\right)\eta_{x,i}^2+4\left(\sigma_{x,i}\eta_{x,i}^2-\sum_{j=1}^n\sigma_{x,i,j}^2\eta_{x,i}\eta_{x,j}\right)+\frac{1}{2}h^\top\nabla^2\sigma_{x,i}h\right)\theta_i\\&-2\Bigg[\sum_{i,j=1}^n\left(2\sigma_{x,i}+\beta_{\mathrm{V}}\right)\left(2\sigma_{x,j}+\beta_{\mathrm{V}}\right)\eta_{x,i}\eta_{x,j}\theta_{x,i,j}^2-2\sum_{i,j,k=1}^n\left(2\sigma_{x,i}+\beta_{\mathrm{V}}\right)\sigma_{x,j,k}^2\theta_{x,i,k}^2\eta_{x,i}\eta_{x,j}\\&+\sum_{i,j,k,l=1}^n\sigma_{x,i,l}^2\sigma_{x,j,k}^2\theta_{x,k,l}^2\eta_{x,i}\eta_{x,j}\Bigg].\end{aligned}\tag{B.37}$$

We claim that the directional Hessian $h^\top\nabla^2\sigma_{x,i}h$ is given by

$$\begin{aligned}&h^\top\nabla^2\sigma_{x,i}h\\&=2\,h^\top A_x^\top\left[E_{ii}(3\Sigma_x-4\operatorname{diag}(\Upsilon_x^{(2)}e_i))E_{ii}+\operatorname{diag}(\Upsilon_xe_i)(4\Upsilon_x-3\mathbb{I})\operatorname{diag}(\Upsilon_xe_i)\right]A_xh.\end{aligned}\tag{B.38}$$

Assuming the claim at the moment we now bound $\left|h^\top \nabla^2 \Psi_x h\right|$. To shorten the notation, we drop the x-dependence of the terms $\sigma_{x,i}, \sigma_{x,i,j}, \theta_{x,i}$ and $\eta_{x,i}$. Since Υ_x is an orthogonal projection matrix, we have

$$\operatorname{diag}(\Upsilon_x e_i)\Upsilon_x \operatorname{diag}(\Upsilon_x e_i) \preceq \operatorname{diag}(\Upsilon_x e_i) \operatorname{diag}(\Upsilon_x e_i).$$

Using this fact and substituting the expression for $h^\top \nabla^2 \sigma_{x,i} h$ from equation (B.38) in equation (B.37), we obtain

$$\begin{aligned}
&\left|h^\top \nabla^2 \Psi_x h\right| \\
&\leq \sum_{i=1}^n \left[3\Big(\sigma_i + \beta_{\mathrm{V}}\Big)\eta_i^2 + 4\big(\sigma_i \eta_i^2 + \sum_{j=1}^n \sigma_{i,j}^2 \eta_i \eta_j\big) + 3\sigma_i \eta_i^2 + 4\sum_{j=1}^n \sigma_{i,j}^2 \eta_i \eta_j + 7\sum_{j=1}^n \sigma_{i,j}^2 \eta_j^2\right]\theta_i \\
&\quad + \left[8\sum_{i,j=1}^n (\sigma_i + \beta_{\mathrm{V}})(\sigma_j + \beta_{\mathrm{V}})\eta_i \eta_j \theta_{i,j}^2 + 8\sum_{i,j,k=1}^n (\sigma_i + \beta_{\mathrm{V}})\sigma_{j,k}^2 \theta_{i,k}^2 \eta_i \eta_j + 2\sum_{i,j,k,l=1}^n \sigma_{i,l}^2 \sigma_{j,k}^2 \theta_{k,l}^2 \eta_i \eta_j\right].
\end{aligned}$$

Rearranging terms, we find that

$$\begin{aligned}
&\left|h^\top \nabla^2 \Psi_x h\right| \\
&\leq \sum_{i=1}^n \left[10\,(\sigma_i + \beta_{\mathrm{V}})\,\eta_i^2 + 8\sum_{j=1}^n \sigma_{i,j}^2 \eta_i \eta_j + 7\sum_{j=1}^n \sigma_{i,j}^2 \eta_j^2\right]\theta_i \\
&\quad + \left[8\sum_{i,j=1}^n (\sigma_i + \beta_{\mathrm{V}})(\sigma_j + \beta_{\mathrm{V}})\eta_i \eta_j \theta_{i,j}^2 + 8\sum_{i,j,k=1}^n (\sigma_i + \beta_{\mathrm{V}})\sigma_{j,k}^2 \theta_{i,k}^2 \eta_i \eta_j + 2\sum_{i,j,k,l=1}^n \sigma_{i,l}^2 \sigma_{j,k}^2 \theta_{k,l}^2 \eta_i \eta_j\right] \\
&\overset{(i)}{\leq} \sum_{i=1}^n \left[10\,(\sigma_i + \beta_{\mathrm{V}})\,\eta_i^2 + 4\sum_{j=1}^n \sigma_{i,j}^2 \left(\eta_i^2 + \eta_j^2\right) + 7\sum_{j=1}^n \sigma_{i,j}^2 \eta_j^2\right]\theta_i \\
&\quad + \left[4\sum_{i,j=1}^n (\sigma_i + \beta_{\mathrm{V}})(\sigma_j + \beta_{\mathrm{V}})\theta_{i,j}^2 (\eta_i^2 + \eta_j^2) + 4\sum_{i,j,k=1}^n (\sigma_i + \beta_{\mathrm{V}})\sigma_{j,k}^2 \theta_{i,k}^2 (\eta_i^2 + \eta_j^2) + \sum_{i,j,k,l=1}^n \sigma_{i,l}^2 \sigma_{j,k}^2 \theta_{k,l}^2 (\eta_i^2 + \eta_j^2)\right]
\end{aligned}$$

where in step (i) we have used the AM-GM inequality. Simplifying further, we obtain

$$\begin{aligned}
&\left|h^\top \nabla^2 \Psi_y h\right| \\
&\leq \sum_{i=1}^n \left[14\,(\sigma_i + \beta_{\mathrm{V}})\,\eta_i^2 + 11\sum_{j=1}^n \sigma_{i,j}^2 \eta_j^2\right]\theta_i + \left[\sum_{i=1}^n 12\,(\sigma_i + \beta_{\mathrm{V}})\,\theta_i \eta_i^2 + \sum_{i,j=1}^n 6\sigma_{i,j}^2 \theta_i \eta_j^2\right] \\
&= 26\sum_{i=1}^n (\sigma_i + \beta_{\mathrm{V}})\,\theta_i \eta_i^2 + 17\sum_{i,j=1}^n \sigma_{i,j}^2 \theta_i \eta_j^2.
\end{aligned}$$

Dividing both sides by two completes the proof.

Proof of claim (B.38)**:** In order to compute the directional Hessian of $x \mapsto \sigma_{x,i}$, we need to track the second order terms in equations (B.32a)-(B.32c). Collecting the

second order terms (denoted by $\sigma_h^{(2)}$) in the expansion of $\sigma_{x+h,i} - \sigma_{x,i}$, we obtain

$$\begin{aligned}\sigma_h^{(2)} = 3\,\frac{a_i^\top H_x^{-1} a_i}{s_{x,i}^2}\frac{(a_i^\top h)^2}{s_{x,i}^2} &- 4\,\frac{a_i^\top H_x^{-1}\left(\sum_{j=1}^n \frac{a_j a_j^\top}{s_{x,j}^2}\frac{a_j^\top h}{s_{x,j}}\right) H_x^{-1} a_i}{s_{x,i}^2}\frac{a_i^\top h}{s_{x,i}} \\ &- 3\,\frac{a_i^\top H_x^{-1}\left(\sum_{j=1}^n \frac{a_j a_j^\top}{s_{x,j}^2}\frac{(a_j^\top h)^2}{s_{x,j}^2}\right) H_x^{-1} a_i}{s_{x,i}^2} \\ &+ 4\,\frac{a_i^\top H_x^{-1}\left(\sum_{j=1}^n \frac{a_j a_j^\top}{s_{x,j}^2}\frac{a_j^\top h}{s_{x,j}}\right) H_x^{-1}\left(\sum_{l=1}^n \frac{a_l a_l^\top}{s_{x,l}^2}\frac{a_l^\top h}{s_{x,l}}\right) a_i}{s_{x,i}^2}.\end{aligned}$$

We simply each term on the RHS one by one. Simplifying the first term, we obtain

$$3\,\frac{a_i^\top H_x^{-1} a_i}{s_{x,i}^2}\frac{(a_i^\top h)^2}{s_{x,i}^2} = 3\,\sigma_{x,i}\eta_{x,i}^2 = h^\top 3\,A_x^\top E_{ii}\Sigma_x E_{ii} A_x\,h.$$

For the second term, we have

$$\begin{aligned}4\,\frac{a_i^\top H_x^{-1}\left(\sum_{j=1}^n \frac{a_j a_j^\top}{s_{x,j}^2}\frac{a_j^\top h}{s_{x,j}}\right) H_x^{-1} a_i}{s_{x,i}^2}\frac{a_i^\top h}{s_{x,i}} &= 4\,\eta_{x,i}\sum_{j=1}^n \sigma_{x,i,j}^2\,\eta_{x,j} \\ &= 4\,h^\top A_x^\top E_{ii}\,\mathrm{diag}\left(\Upsilon_x^{(2)} e_i\right) E_{ii} A_x h.\end{aligned}$$

The third term can be simplified as follows:

$$\begin{aligned}3\,\frac{a_i^\top H_x^{-1}\left(\sum_{j=1}^n \frac{a_j a_j^\top}{s_{x,j}^2}\frac{(a_j^\top h)^2}{s_{x,j}^2}\right) H_x^{-1} a_i}{s_{x,i}^2} &= 3\sum_{j=1}^n \sigma_{x,i,j}^2\eta_{x,j}^2 \\ &= 3\,h^\top A_x^\top\,\mathrm{diag}\left(\Upsilon_x e_i\right)\mathrm{diag}\left(\Upsilon_x e_i\right) A_x h\end{aligned}$$

For the last term, we find that

$$\begin{aligned}&4\,\frac{a_i^\top H_x^{-1}\left(\sum_{j=1}^n \frac{a_j a_j^\top}{s_{x,j}^2}\frac{a_j^\top h}{s_{x,j}}\right) H_x^{-1}\left(\sum_{l=1}^n \frac{a_l a_l^\top}{s_{x,l}^2}\frac{a_l^\top h}{s_{x,l}}\right) a_i}{s_{x,i}^2} \\ &= 4\sum_{j,l=1}^n \sigma_{x,i,j}\,\sigma_{x,j,l}\,\sigma_{x,l,i}\,\eta_{x,j}\,\eta_{x,l} \\ &= 4\,h^\top A_x^\top\,\mathrm{diag}\left(\Upsilon_x e_i\right)\Upsilon_x\,\mathrm{diag}\left(\Upsilon_x e_i\right) A_x h.\end{aligned}$$

Putting together the pieces yields the expression (B.38).

B.4 Analysis of the John walk

We recap the key ideas of the John walk for convenience. We have designed a new proposal distribution by making use of an *optimal set of weights* to define the new covariance structure for the Gaussian proposals, where optimality is defined with respect to the convex program defined below (B.39). The optimality condition is closely related to the problem of finding the largest ellipsoid at any interior point of the polytope, such that the ellipsoid is contained within the polytope. This problem of finding the largest ellipsoid was first studied by [92] who showed that each convex body in $\mathbb{R}^d$ contains a unique ellipsoid of maximal volume. More recently, [107] make use of approximate John Ellipsoids to improve the convergence rate of interior point methods for linear programming. We refer the readers to their paper for more discussion about the use of John Ellipsoids for optimization problems. In this work, we make use of these ellipsoids for designing sampling algorithms with better theoretical bounds on the mixing times.

The vector $\zeta_x = (\zeta_{x,1}, \ldots, \zeta_{x,n})^\top$ defined in the John walk's inverse covariance matrix (4.8) is computed by solving the following optimization problem:

$$\zeta_x = \arg\min_{w \in \mathbb{R}^n} c_x(w) := \sum_{i=1}^n w_i - \frac{1}{\alpha_{\mathrm{J}}} \log\det\left(A^\top S_x^{-1} W^{\alpha_{\mathrm{J}}} S_x^{-1} A\right) - \beta_{\mathrm{J}} \sum_{i=1}^n \log w_i, \quad \text{(B.39)}$$

where the parameters $\alpha_{\mathrm{J}}, \beta_{\mathrm{J}}$ are given by

$$\alpha_{\mathrm{J}} = 1 - \frac{1}{\log_2(2n/d)} \quad \text{and} \quad \beta_{\mathrm{J}} = \frac{d}{2n},$$

and W denotes an $n \times n$ diagonal matrix with $W_{ii} = w_i$ for each $i \in [n]$. In particular, for our proposal the inverse covariance matrix is proportional to J_x, where

$$J_x = \sum_{i=1}^n \zeta_{x,i} \frac{a_i a_i^\top}{(b_i - a_i^\top x)^2}. \quad \text{(B.40)}$$

where $\kappa := \kappa_{n,d} = \log_2(2n/d) = (1 - \alpha_{\mathrm{J}})^{-1}$.

Recall that for John walk with parameter $\frac{r}{d^{3/4}\kappa^2}$, the proposals at state x are drawn from the multivariate Gaussian distribution given by $\mathcal{N}\left(x, \frac{r^2}{d^{3/2}\kappa^4} J_x^{-1}\right)$, which we denote by $\mathcal{P}_x^{\mathrm{J}}$. In particular, the proposal density at point $x \in \mathrm{int}(\mathcal{K})$ is given by

$$\rho_x(z) := \rho(x,z) = \sqrt{\det J_x} \left(\frac{\kappa^4 d^{3/2}}{2\pi r^2}\right)^{d/2} \exp\left(-\frac{\kappa^4 d^{3/2}}{2r^2} (z-x)^\top J_x (z-x)\right). \quad \text{(B.41)}$$

Here we restate our result for the mixing time of the John walk. Scountertheorem1

Theorem 14. *Let μ_0 be any distribution that is M-warm with respect to Π^* and let $n < \exp(\sqrt{d})$. Then for any $\delta \in (0,1]$, the John walk with parameter $r_{John} = 10^{-5}$ satisfies*

$$d_{TV}\left(\mathcal{T}^k_{John(r)}(\mu_0), \Pi^*\right) \le \delta \qquad \textit{for all } k \ge C\, d^{2.5} \log_2^4\left(\frac{2n}{d}\right) \log\left(\frac{\sqrt{M}}{\delta}\right).$$

B.4.1 Auxiliary results

We begin by proving basic properties of the weights ζ_x which are used throughout the chapter. For $x \in \text{int}\,(\mathcal{K})\,, w \in \mathbb{R}^n_{++}$, define the projection matrix $\Upsilon_{x,w}$ as follows

$$\Upsilon_{x,w} = W^{\alpha/2} A_x (A_x^\top W^\alpha A_x)^{-1} A_x^\top W^{\alpha/2}, \tag{B.42}$$

where $A_x = S_x^{-1} A$ and W is the $n \times n$ diagonal matrix with i-th diagonal entry given by w_i. Also, let

$$\sigma_{x,i} := (\Upsilon_{x,\zeta_x})_{ii} \quad \text{for } x \in \text{int}\,(\mathcal{K}) \text{ and } i \in [n]. \tag{B.43}$$

Define the *John slack sensitivity* θ_x^{J} as

$$\theta_x := \theta_x^{\text{J}} := \left(\frac{a_1^\top J_x^{-1} a_1}{s_{x,1}^2}, \ldots, \frac{a_n^\top J_x^{-1} a_n}{s_{x,n}^2} \right)^\top \quad \text{for all } x \in \text{int}\,(\mathcal{K})\,. \tag{B.44}$$

Further, for any $x \in \text{int}\,(\mathcal{K})$, define the *John local norm at* x as

$$\|\cdot\|_{J_x} : v \mapsto \left\| J_x^{1/2} v \right\|_2 = \sqrt{\sum_{i=1}^n \zeta_{x,i} \frac{(a_i^\top v)^2}{s_{x,i}^2}}. \tag{B.45}$$

We now collect some basic properties of the weights ζ_x and the local sensitivity θ_x and restate parts of Lemma 11 for clarity here.

Lemma 28. *For any $x \in \text{int}\,(\mathcal{K})$, the following properties are true:*

(a) (Implicit weight formula) $\zeta_{x,i} = \sigma_{x,i} + \beta_J$ for all $i \in [n]$,

(b) (Uniformity) $\zeta_{x,i} \in [\beta_J, 1 + \beta_J]$ for all $i \in [n]$,

(c) (Total size) $\sum_{i=1}^n \zeta_{x,i} = 3d/2$, and

(d) (Slack sensitivity) $\theta_{x,i} \in [0, 4]$ for all $i \in [n]$.

Lemma 28 follows from Lemmas 14 and 15 by [107] and thereby we omit its proof.

Next, we state a key lemma that is crucial for proving the convergence rate of John walk. In this lemma, we provide bounds on difference in total variation norm between the proposal distributions of two nearby points.

Lemma 29. *There exists a continuous non-decreasing function $h : [0, 1/30] \to \mathbb{R}_+$ with $h(1/30) \geq 10^{-5}$, such that for any $\epsilon \in (0, 1/30]$, the John walk with $r \in [0, h(\epsilon)]$ satisfies*

$$d_{TV}\left(\mathcal{P}_x^J, \mathcal{P}_y^J\right) \leq \epsilon, \quad \textit{for all } x, y \in \text{int}\,(\mathcal{K}) \textit{ such that } \|x - y\|_{J_x} \leq \frac{\epsilon r}{2\kappa^2 d^{3/4}}\,, \textit{ and} \tag{B.46a}$$

$$d_{TV}\left(\mathcal{T}_{John(r)}(\delta_x), \mathcal{P}_x^J\right) \leq 5\epsilon, \quad \textit{for all } x \in \text{int}\,(\mathcal{K}). \tag{B.46b}$$

See Section B.4.3 for its proof.

With these lemmas in hand, we are now ready to prove Theorem 14.

B.4.2 Proof of Theorem 14

The proof is similar to the proof of Theorem 1, and relies on the Lovász's Lemma. Here onwards, we use the following simplified notation

$$\mathcal{T}_x = \mathcal{T}_{\text{John}(r)}(\delta_x), \mathcal{P}_x = \mathcal{P}_x^{\text{J}} \text{ and } \|\cdot\|_x = \|\cdot\|_{J_x}.$$

In order to invoke Lovász's Lemma, we need to show that for any two points $x, y \in \text{int}(\mathcal{K})$ with small cross-ratio $d_{\mathcal{K}}(x,y)$, the TV-distance $d_{\text{TV}}(\mathcal{T}_x, \mathcal{T}_y)$ is also small.

We proceed with the proof in two steps: (A) first, we relate the cross-ratio $d_{\mathcal{K}}(x,y)$ to the John local norm of $x-y$ at x, and (B) we then use Lemma 29 to show that if $x, y \in \text{int}(\mathcal{K})$ are close in the John local-norm, then the transition kernels $\mathcal{T}_x$ and $\mathcal{T}_y$ are close in TV-distance.

Step (A): We claim that for all $x, y \in \text{int}(\mathcal{K})$, the cross-ratio can be lower bounded as

$$d_{\mathcal{K}}(x,y) \geq \frac{1}{\sqrt{3d/2}} \|x-y\|_x. \tag{B.47}$$

From the arguments in the proof of Theorem 1 (proof for the Vaidya Walk), we have

$$d_{\mathcal{K}}(x,y) \geq \max_{i\in[n]} \left| \frac{a_i^\top (x-y)}{s_{x,i}} \right|. \tag{B.48}$$

Using the fact that maximum of a set of non-negative numbers is greater than the weighted mean of the numbers and Lemma 28, we find that

$$d_{\mathcal{K}}(x,y) \geq \sqrt{\frac{1}{\sum_{i=1}^n \zeta_{x,i}} \sum_{i=1}^n \zeta_{x,i} \frac{(a_i^\top (x-y))^2}{s_{x,i}^2}} = \frac{\|x-y\|_x}{\sqrt{3d/2}},$$

thereby proving the claim (B.47).

Step (B): By the triangle inequality, we have

$$d_{\text{TV}}(\mathcal{T}_x, \mathcal{T}_y) \leq d_{\text{TV}}(\mathcal{T}_x, \mathcal{P}_x) + d_{\text{TV}}(\mathcal{P}_x, \mathcal{P}_y) + d_{\text{TV}}(\mathcal{P}_y, \mathcal{T}_y).$$

Using Lemma 29, we obtain that

$$d_{\text{TV}}(\mathcal{T}_x, \mathcal{T}_y) \leq 11\epsilon, \quad \forall x, y \in \text{int}(\mathcal{K}) \text{ such that } \|x-y\|_x \leq \frac{\epsilon r}{2\kappa^2 d^{3/4}}.$$

Consequently, the John walk satisfies the assumptions of Lovász's Lemma with

$$\Delta := \frac{1}{\sqrt{3d/2}} \cdot \frac{\epsilon r}{2\kappa^2 d^{3/4}} \quad \text{and} \quad \rho := 1 - 11\epsilon.$$

Plugging in $\epsilon = 1/30$, $r = 10^{-5}$, we obtain the claimed upper bound of $O\left(\kappa^4 d^{5/2}\right)$ on the mixing time of the random walk.

B.4.3 Proof of Lemma 29

We prove the lemma for the following function,

$$h(\epsilon) = \min\left\{ \frac{1}{25\sqrt{1+\sqrt{2}\log(4/\epsilon)}}, \frac{\epsilon}{(2\sqrt{32}\chi_{1,\epsilon})}, \sqrt{\frac{\epsilon}{386\sqrt{24}\chi_{2,\epsilon}}}, \frac{\epsilon}{5\sqrt{60}\chi_{3,\epsilon}}, \right.$$
$$\left. \sqrt{\frac{\epsilon}{8\sqrt{1680}\chi_{4,\epsilon}}}, \sqrt{\frac{\epsilon}{40\left(\chi_{2,\epsilon}\chi_{6,\epsilon}\sqrt{24}\sqrt{15120}\right)^{1/2}}}, \sqrt{\frac{\epsilon}{204800\chi_{2,\epsilon}\sqrt{24}\log(32/\epsilon)}} \right\}.$$

where $\chi_{1,\epsilon} = \log(2/\epsilon)$and $\chi_{k,\epsilon} = (2e/k \cdot \log(16/\epsilon))^{k/2}$ for $k = 2, 3, 4$ and 6. A numerical calculation shows that $h(1/30) \geq 10^{-5}$.

We now prove the two parts (B.46a) (B.46b) of the Lemma separately.

Proof of claim (B.46a)

Applying Pinsker's inequality, and plugging in the closed formed expression for the KL divergence between two Gaussian distributions we find that

$$\begin{aligned} d_{\mathrm{TV}}(\mathcal{P}_x, \mathcal{P}_y)^2 &\leq 2\mathrm{KL}(\mathcal{P}_y \parallel \mathcal{P}_x) \\ &= \mathrm{trace}(J_x^{-1/2} J_y J_x^{-1/2}) - d - \log\det(J_x^{-1/2} J_y J_x^{-1/2}) + \frac{\kappa^4 d^{3/2}}{r^2} \|x-y\|_x^2 \\ &= \sum_{i=1}^{d} \left(\lambda_i - 1 + \log\frac{1}{\lambda_i}\right) + \frac{\kappa^4 d^{3/2}}{r^2} \|x-y\|_x^2, \end{aligned} \tag{B.49}$$

where $\lambda_1, \ldots, \lambda_d > 0$ denote the eigenvalues of the matrix $J_x^{-1/2} J_y J_x^{-1/2}$. To bound the expression (B.49), we make use of the following lemma:

Lemma 30. *For any scalar $t \in [0, 1/64]$ and pair of points $x, y \in \mathrm{int}(\mathcal{K})$ such that $\|x-y\|_x \leq t/\kappa^2$, we have*

$$\left(1 - 48t + 4t^2\right)\mathbb{I}_d \preceq J_x^{-1/2} J_y J_x^{-1/2} \preceq \left(1 + 48t + 4t^2\right),$$

where $\preceq$ denotes ordering in the PSD cone and $\mathbb{I}_d$ denotes the d-dimensional identity matrix.

See Section B.6 for the proof of this lemma.

For $\epsilon \in (0, 1/30]$ and $r = 10^{-5}$, we have $t = \epsilon r/(2d^{3/4}) \leq 1/64$, whence the eigenvalues $\{\lambda_i, i \in [d]\}$ can be sandwiched as

$$1 - \frac{24\epsilon r}{d^{3/4}} + \frac{\epsilon^2 r^2}{d^{3/2}} \leq \lambda_i \leq 1 + \frac{24\epsilon r}{d^{3/4}} + \frac{\epsilon^2 r^2}{d^{3/2}} \quad \text{for all } i \in d. \tag{B.50}$$

We are now ready to bound the TV distance between $\mathcal{P}_x$ and $\mathcal{P}_y$. Using the bound (B.49) and the inequality $\log\omega \leq \omega - 1$, valid for $\omega > 0$, we obtain

$$d_{\mathrm{TV}}\left(\mathcal{P}_x, \mathcal{P}_y\right)^2 \leq \sum_{i=1}^{d}\left(\lambda_i - 2 + \frac{1}{\lambda_i}\right) + \frac{\kappa^4 d^{3/2}}{r^2}\left\|x-y\right\|_x^2.$$

Using the assumption that $\|x-y\|_x \leq \epsilon r/\left(2\kappa^2 d^{3/4}\right)$, and plugging in the bounds (B.50) for the eigenvalues $\{\lambda_i, i \in [d]\}$, we find that

$$\sum_{i=1}^{d}\left(\lambda_i - 2 + \frac{1}{\lambda_i}\right) + \frac{\kappa^4 d^{3/2}}{r^2}\left\|x-y\right\|_x^2 \leq \frac{2000\epsilon^2 r^2}{\sqrt{d}} + \frac{\epsilon^2}{4}.$$

In asserting this inequality, we have used the facts that

$$\frac{1}{1-24\omega+\omega^2} \leq 1 + 24\omega + 1000\omega^2, \quad \text{and} \quad \frac{1}{1+24\omega+\omega^2} \leq 1 - 24\omega + 1000\omega^2,$$

for all $\omega \in \left[0, \frac{1}{100}\right]$. Note that for any $r \in [0, 1/100]$, we have that $2000r^2/\sqrt{d} \leq 1/2$. Putting the pieces together yields $d_{\mathrm{TV}}\left(\mathcal{P}_x, \mathcal{P}_y\right) \leq \epsilon$, as claimed.

Proof of claim (B.46b)

We have

$$d_{\mathrm{TV}}\left(\mathcal{P}_x, \mathcal{T}_x\right) \leq \underbrace{\frac{3}{2}\mathcal{P}_x(\mathcal{K}^c)}_{=:\, S_1} + \underbrace{1 - \mathbb{E}_{z\sim\mathcal{P}_x}\left[\min\left\{1, \frac{\rho_z(x)}{\rho_x(z)}\right\}\right]}_{=:\, S_2}, \tag{B.51}$$

where $\mathcal{K}^c$ denotes the complement of $\mathcal{K}$. We now show that $S_1 \leq \epsilon$ and $S_2 \leq 4\epsilon$, from which the claim follows.

Bounding the term S_1: Note that for $z \sim \mathcal{N}(x, \frac{r^2}{\kappa^2 d^{3/2}} J_x^{-1})$, we can write

$$z \overset{d}{=} x + \frac{r}{\kappa d^{3/4}} J_x^{-1/2}\xi, \tag{B.52}$$

where $\xi \sim \mathcal{N}\left(0, \mathbb{I}_d\right)$ and $\overset{d}{=}$ denotes equality in distribution. Using equation (B.52) and definition (B.44) of $\theta_{x,i}$, we obtain the bound

$$\frac{\left(a_i^\top (z-x)\right)^2}{s_{x,i}^2} = \frac{r^2}{\kappa^2 d^{3/2}}\left[\frac{a_i^\top J_x^{-1/2}\xi}{s_{x,i}}\right]^2 \overset{(i)}{\leq} \frac{r^2}{\kappa^2 d^{3/2}}\theta_{x,i}\left\|\xi\right\|_2^2 \overset{(ii)}{\leq} \frac{4r^2}{d}\left\|\xi\right\|_2^2, \tag{B.53}$$

where step (i) follows from Cauchy-Schwarz inequality, and step (ii) from part (d) of Lemma 28. Define the events

$$\mathcal{E} := \left\{\frac{r^2}{d}\left\|\xi\right\|_2^2 < \frac{1}{4}\right\} \quad \text{and} \quad \mathcal{E}' := \left\{z \in \mathrm{int}\left(\mathcal{K}\right)\right\}.$$

Inequality (B.53) implies that $\mathcal{E} \subseteq \mathcal{E}'$ and hence $\mathbb{P}[\mathcal{E}'] \geq \mathbb{P}[\mathcal{E}]$. Using a standard Gaussian tail bound and noting that $r \leq \frac{1/2}{1+\sqrt{2/d \log(2/\epsilon)}}$, we obtain $\mathbb{P}[\mathcal{E}] \geq 1-\epsilon/2$ and whence $\mathbb{P}[\mathcal{E}'] \geq 1-\epsilon/2$. Thus, we have shown that $\mathbb{P}[z \notin \mathcal{K}] \leq \epsilon/2$ which implies that $S_1 \leq \epsilon$.

Bounding the term S_2: By Markov's inequality, we have

$$\mathbb{E}_{z\sim\mathcal{P}_x}\left[\min\left\{1, \frac{\rho_z(x)}{\rho_x(z)}\right\}\right] \geq \alpha \mathbb{P}[\rho_z(x) \geq \alpha \rho_x(z)] \quad \text{for all } \alpha \in (0,1]. \tag{B.54}$$

By definition (B.41) of ρ_x, we obtain

$$\frac{\rho_z(x)}{\rho_x(z)} = \exp\left(-\frac{d^{3/2}\kappa^4}{2r^2}\left(\|z-x\|_z^2 - \|z-x\|_x^2\right) + \frac{1}{2}\left(\log\det J_z - \log\det J_x\right)\right).$$

The following lemma provides us with useful bounds on the two terms in this expression, valid for any $x \in \text{int}(\mathcal{K})$.

Lemma 31. *For any $\epsilon \in (0, \frac{1}{4}]$ and $r \in (0, h(\epsilon)]$, we have*

$$\mathbb{P}_{z\sim\mathcal{P}_x}\left[\frac{1}{2}\log\det J_z - \frac{1}{2}\log\det J_x \geq -\epsilon\right] \geq 1-\epsilon, \quad \textit{and} \tag{B.55a}$$

$$\mathbb{P}_{z\sim\mathcal{P}_x}\left[\|z-x\|_z^2 - \|z-x\|_x^2 \leq 2\epsilon\frac{r^2}{\kappa^4 d^{3/2}}\right] \geq 1-\epsilon. \tag{B.55b}$$

We provide the of this lemma in Section B.7.

Using Lemma 31, we now complete the proof of the Theorem 14. For $r \leq h(\epsilon)$, we obtain

$$\frac{\rho_z(x)}{\rho_x(z)} \geq \exp(-2\epsilon) \geq 1-2\epsilon$$

with probability at least $1-2\epsilon$. Substituting $\alpha = 1-2\epsilon$ in inequality (B.54) yields that $S_2 \leq 4\epsilon$, as claimed.

B.5 Technical Lemmas for the John walk

We begin by summarizing a few key properties of various terms involved in our analysis.

Let $\Sigma_{x,w}$ be an $n \times n$ diagonal matrix defined as

$$\Sigma_{x,w} = \text{diag}(\sigma_{x,w,i}, \ldots, \sigma_{x,w,n}) \text{ where } \sigma_{x,\zeta_x,w,i} = (\Upsilon_{x,w})_{ii}, i \in [n]. \tag{B.56a}$$

Let $\Upsilon_{x,w}^{(2)}$ denote the hadamard product of $\Upsilon_{x,w}$ with itself. Further define

$$\Lambda_{x,w} := \Sigma_{x,w} - \Upsilon_{x,w}^{(2)}. \tag{B.56b}$$

[107] proved that the weight vector ζ_x is the unique solution of the following fixed point equation:

$$w_i = \sigma_{x,w,i} + \beta_{\mathrm{J}}, i \in [n]. \tag{B.57a}$$

To simplify notation, we use the following shorthands:

$$\sigma_x = \sigma_{x,\zeta_x}, \quad \Upsilon_x = \Upsilon_{x,\zeta_x}, \quad \Upsilon_x^{(2)} = \Upsilon_{x,\zeta_x}^{(2)}, \quad \Sigma_x = \Sigma_{x,\zeta_x}, \quad \Lambda_x = \Lambda_{x,\zeta_x}. \tag{B.57b}$$

Thus, we have the following relation:

$$\zeta_x = \sigma_{x,\zeta_x} + \beta_{\mathrm{J}}\mathbf{1} = \sigma_x + \beta_{\mathrm{J}}\mathbf{1}. \tag{B.57c}$$

B.5.1 Deterministic expressions and bounds

We now collect some properties of various terms defined above.

Lemma 32. *For any $x \in \operatorname{int}(\mathcal{K})$, the following properties hold:*

(a) $\sigma_{x,i} = \sum_{j=1}^n \sigma_{x,i,j}^2 = \sum_{j,k=1}^n \sigma_{x,i,j}\sigma_{x,j,k}\sigma_{x,k,i}$ *for each* $i \in [n]$,

(b) $\Sigma_x \succeq \Upsilon_x^{(2)}$,

(c) $\sum_{i=1}^n \zeta_{x,i}\theta_{x,i} = d$,

(d) $\theta_{x,i} = \sum_{j=1}^n \zeta_{x,i}\theta_{x,i,j}^2$, *for each* $i \in [n]$,

(e) $\theta_x^\top \Sigma_x \theta_x = \sum_{i=1}^n \theta_{x,i}^2 \zeta_{x,i} \leq 4d$, *and*

(f) $\beta_J \nabla^2 \mathcal{F}_x \preceq J_x \preceq (1+\beta_J)\nabla^2 \mathcal{F}_x$.

The proof is based on the ideas similar to Lemma 5 in the proof of the Vaidya walk and is thereby omitted.

The next lemma relates the change in *slackness* $s_{x,i} = b_i - a_i^\top x$ to the John-local norm at x.

Lemma 33. *For all $x, y \in \operatorname{int}(\mathcal{K})$, we have*

$$\max_{i\in[n]} \left| 1 - \frac{s_{y,i}}{s_{x,i}} \right| \leq 2 \|x-y\|_x .$$

Proof. For any pair $x, y \in \operatorname{int}(\mathcal{K})$ and index $i \in [n]$, we have

$$\left(a_i^\top (x-y)\right)^2 \overset{(i)}{\leq} \|J_x^{-\frac{1}{2}} a_i\|_2^2 \, \|J_x^{\frac{1}{2}}(x-y)\|_2^2 = \theta_{x,i} s_{x,i}^2 \, \|x-y\|_x^2 \overset{(ii)}{\leq} 4 s_{x,i}^2 \, \|x-y\|_x^2 ,$$

where step (i) follows from the Cauchy-Schwarz inequality, and step (ii) uses the bound $\theta_{x,i}$ from Lemma 28(d). Noting the fact that $a_i^\top(x-y) = s_{y,i} - s_{x,i}$, the claim follows after simple algebra. □

We now state various expressions and bounds for the first and second order derivatives of the different terms. To lighten notation, we introduce some shorthand notation. For any $y \in \operatorname{int}(\mathcal{K})$ and $h \in \mathbb{R}^d$, define the following terms:

$$d_{y,i} = \frac{a_i^\top h}{s_{y,i}},\ i \in [n] \qquad D_y = \operatorname{diag}(d_{y,1}, \ldots, d_{y,n}), \tag{B.58a}$$

$$f_{y,i} = \frac{\nabla \zeta_{y,i}^\top h}{\zeta_{y,i}},\ i \in [n] \qquad F_y = \operatorname{diag}(f_{y,1}, \ldots, f_{y,n}), \tag{B.58b}$$

$$\ell_{y,i} = \frac{1}{2} h^\top \nabla^2 \zeta_{y,i} h / \zeta_{y,i},\ i \in [n] \qquad L_y = \operatorname{diag}(\ell_{y,1}, \ldots, \ell_{y,n}), \tag{B.58c}$$

$$\rho_y := (G_y - \alpha \Lambda_y) \begin{bmatrix} \ell_{y,1} \\ \vdots \\ \ell_{y,n} \end{bmatrix}, \tag{B.58d}$$

where for brevity in our notation we have omitted the dependence on h. The choice of h is specified as per the context. Further, we define for each $x \in \operatorname{int}(\mathcal{K})$ and $i \in [n]$

$$\varphi_{x,i} := \frac{\zeta_{x,i}}{s_{x,i}^2}, \qquad \text{and} \qquad \Psi_x := \frac{1}{2} \log\det J_x, \tag{B.59}$$

$$\hat{a}_{x,i} := \frac{J_x^{-1/2} a_{x,i}}{s_{x,i}^2}, \qquad \text{and} \qquad \hat{b}_{x,i} := J_x^{-1/2} A_x \Lambda_x \left(G_x - \alpha \Lambda_x\right)^{-1} e_i. \tag{B.60}$$

Next, we state expressions for gradients of ζ, φ and Ψ and bounds for directional Hessian of σ, φ and Ψ which are used in various Taylor series expansions and bounds in our proof.

Lemma 34 (Calculus)**.** *For any $y \in \operatorname{int}(\mathcal{K})$ and $h \in \mathbb{R}^n$, the following relations hold;*

(a) Gradient of ζ: $(f_{y,1}, \ldots, f_{y,n})^\top = 2 \left(G_y - \alpha \Lambda_y\right)^{-1} \Lambda_y A_y h$*;*

(b) Hessian of ζ:

$$\|\rho_y\|_1 \leq 56 \kappa^2 \sum_{i=1}^n \zeta_{y,i} d_{y,i}^2. \tag{B.61}$$

(c) Gradient of Ψ: $\nabla \Psi^\top h = \theta_y^\top G_y \left(\mathbb{I}_n + \left(G_y - \alpha \Lambda_y\right)^{-1} \Lambda_y\right) A_y h$*.*

(d) Gradient of φ: $\nabla \varphi_{y,i}^\top h = \varphi_{y,i} \left(2 d_{y,i} + f_{y,i}\right)$*.*

(e) Bound on $\nabla^2 \Psi$: $\frac{1}{2} \left| h^\top (\nabla^2 \Psi) h \right| \leq \frac{1}{2} \left[\sum_{i=1}^n \zeta_{y,i}\, \theta_{y,i} \left[9\, d_{y,i}^2 + 4 f_{y,i}^2 \right] + \left| \sum_{i=1}^n \zeta_{y,i}\, \theta_{y,i} \ell_{y,i} \right| \right]$

(f) Bound on $\nabla^2 \varphi$:

$$\left| \sum_{i=1}^n d_{y,i}^2 s_{y,i}^2 \frac{1}{2} h^\top \nabla^2 \varphi_{y,i} h \right| \leq 3 \sum_{i=1}^n \zeta_{y,i} d_{y,i}^4 + 2 \left| \sum_{i=1}^n \zeta_{y,i} d_{y,i}^3 f_{y,i} \right| + \left| \sum_{i=1}^n \zeta_{y,i} d_{y,i}^2 \ell_{y,i} \right|.$$

The proof is provided in Section B.8.1.

Next, we state some results that would be useful to provide explicit bounds for various terms like f_y, ℓ_y and ρ_y that appear in the statements of the previous lemma. Note that the following results do not have a corresponding analog in our analysis of the Vaidya walk.

Lemma 35. *For any $c_1, c_2 \geq 0$, $y \in \operatorname{int}(\mathcal{K})$, we have*

$$\left(c_1 \mathbb{I}_n + c_2 \Lambda_y \left(G_y - \alpha \Lambda_y\right)^{-1}\right) G_y \left(c_1 \mathbb{I}_n + c_2 \left(G_y - \alpha \Lambda_y\right)^{-1} \Lambda_y\right) \preceq \left(c_1 + c_2\right)^2 \kappa^2 G_y,$$

where $\preceq$ denotes the ordering in the PSD cone.

Lemma 36. *Let μ_y denote the $n \times n$ matrix $\left(G_y - \alpha \Lambda_y\right)^{-1} G_y$, and let $\mu_{y,i,j}$ denote its ij-th entry. Then for each $i \in [n]$ and $y \in \operatorname{int}(\mathcal{K})$, we have*

$$\mu_{y,i,i} \in [0, \kappa], \quad \text{and,} \tag{B.62a}$$

$$\sum_{j \neq i, j \in [n]} \frac{\mu_{y,i,j}^2}{\zeta_{y,j}} \leq \kappa^3. \tag{B.62b}$$

Corollary 8. *Let $e_i \in \mathbb{R}^n$ denote the unit vector along i-th axis. Then for any $y \in \operatorname{int}(\mathcal{K})$, we have*

$$\left\| G_y \left(G_y - \alpha \Lambda_y\right)^{-1} e_i \right\|_1 \leq 3\sqrt{d}\kappa^{3/2}, \quad \text{for all } i \in [n]. \tag{B.63}$$

Consequently, we also have $\left\| \left(G_y - \alpha \Lambda_y\right)^{-1} G_y \right\|_\infty \leq 3\sqrt{d}\kappa^{3/2}$.

See Section B.8.2, B.8.3 and B.8.4 for the proofs of Lemma 35, Lemma 36 and Corollary 8 respectively.

B.5.2 Tail Bounds

We now collect lemmas that provide us with useful tail bounds.

We start with a result that shows that for a random variable $z \sim \mathcal{P}_x$, the slackness $s_{z,i}$ is close to $s_{x,i}$ with high probability and consequently the weights $\zeta_{z,i}$ are also close to $\zeta_{x,i}$. This result comes in handy for transferring the remainder terms in Taylor expansions to the reference point (around which the series is being expanded).

Lemma 37. *For any point $x \in \operatorname{int}(\mathcal{K})$ and $r \leq \frac{1}{25 \cdot \sqrt{1+\sqrt{2}\log(4/\epsilon)}}$, we have*

$$\mathbb{P}_{z \sim \mathcal{P}_x}\left[\forall i \in [n], \forall v \in \overline{xz},\ \frac{s_{x,i}}{s_{v,i}} \in [0.99, 1.01] \ \text{ and } \frac{\zeta_{x,i}}{\zeta_{v,i}} \in [0.96, 1.04]\right] \geq 1 - \epsilon/4 \tag{B.64a}$$

See Section B.9.1 for the proof of this lemma.

Next, we state high probability results for some Gaussian polynomials. These results are useful to bound various polynomials of the form $\sum_{i=1}^n \zeta_{x,i} d_{x,i}^k$, where $d_{x,i} = a_i^\top (z - x)/s_{x,i}$ and z is drawn from the transition distribution for the John walk at point x.

Lemma 38 (Gaussian moment bounds)**.** *To simplify notations, all subscripts on x are omitted in the following statements. For any $\epsilon \in (0, 1/30]$, define $\chi_k := \chi_{k,\epsilon} = (2e/k \cdot \log(16/\epsilon))^{k/2}$, for $k = 2, 3, 4$ and 6, then we have*

$$\mathbb{P}\left[\sum_{i=1}^{n} \zeta_i \left(\hat{a}_i^\top \xi\right)^2 \le \chi_2 \sqrt{24} d\right] \ge 1 - \frac{\epsilon}{16}, \tag{B.65a}$$

$$\mathbb{P}\left[\sum_{i=1}^{n} \zeta_i \left(\hat{a}_i^\top \xi\right)^3 \le \chi_3 \sqrt{60} d^{1/2}\right] \ge 1 - \frac{\epsilon}{16}, \tag{B.65b}$$

$$\mathbb{P}\left[\sum_{i=1}^{n} \zeta_i \left(\hat{a}_i^\top \xi\right)^2 \left(\hat{b}_i^\top \xi\right) \le \chi_3 \sqrt{240} \kappa d^{1/2}\right] \ge 1 - \frac{\epsilon}{16}, \tag{B.65c}$$

$$\mathbb{P}\left[\sum_{i=1}^{n} \zeta_i \left(\hat{a}_i^\top \xi\right)^4 \le \chi_4 \sqrt{1680} d\right] \ge 1 - \frac{\epsilon}{16}, \tag{B.65d}$$

$$\mathbb{P}\left[\sum_{i=1}^{n} \zeta_i \left(\hat{a}_i^\top \xi\right)^6 \le \chi_6 \sqrt{15120} d\right] \ge 1 - \frac{\epsilon}{16}. \tag{B.65e}$$

See Section B.9.2 for the proof.

B.6 Proof of Lemma 30

As a direct consequence of Lemma 33, for any $x, y \in \operatorname{int}(\mathcal{K})$ such that $\|x - y\|_x \le t/\kappa^2$, we have

$$\max_{i \in [n]} \left| 1 - \frac{s_{y,i}}{s_{x,i}} \right| \le \frac{2t}{\kappa^2}. \tag{B.66}$$

Bounding the terms in $\nabla^2 \mathcal{F}_x$ one by one, we obtain

$$\left(1 - \frac{2t}{\kappa^2}\right)^2 \nabla^2 \mathcal{F}_y \preceq \nabla^2 \mathcal{F}_x \preceq \left(1 + \frac{2t}{\kappa^2}\right)^2 \nabla^2 \mathcal{F}_y.$$

We claim that

$$\|\log \zeta_y - \log \zeta_x\|_\infty \le 16t. \tag{B.67}$$

Assuming the claim as given at the moment, we now complete the proof. Putting the result (B.67) in matrix form, we obtain that $\exp(-16t)\, \mathbb{I}_n \preceq G_x^{-1} G_y \preceq \exp(16t)\, \mathbb{I}_n$, and hence

$$\exp(-16t)\, \zeta_{x,i} \le \zeta_{y,i} \le \exp(16t)\, \zeta_{x,i}. \tag{B.68}$$

Consequently, using the definition of J_x we have,

$$\underbrace{\left(1-\frac{2t}{\kappa^2}\right)^2 \exp\left(-16t\right)}_{\omega_\ell} J_x \preceq J_y \preceq \underbrace{\left(1+\frac{2t}{\kappa^2}\right)^2 \exp\left(16t\right)}_{\omega_u} J_y.$$

Letting $\omega = 2t$, we obtain

$$\omega_\ell \geq (1-\omega)^2 \cdot \exp\left(-8\omega\right) \overset{(i)}{\geq} 1 - 24\omega + \omega^2, \quad \text{and}$$
$$\omega_u \leq (1+\omega)^2 \cdot \exp\left(8\omega\right) \overset{(ii)}{\leq} 1 + 24\omega + \omega^2,$$

where inequalities (i) and (ii) hold since $\omega \leq 1/24$. Putting the pieces together, we find that

$$\left(1 - 48t + 4t^2\right) J_x \preceq J_y \preceq \left(1 - 48t + 4t^2\right) J_x$$

for $t \in [0, 1/48]$.

Now, we return to the proof of our earlier claim (B.67). We use an argument based on the continuity of the function $x \mapsto \log \zeta_x$. (Such an argument appeared in a similar scenario in [107].) For $\lambda \in [0,1]$, define $u_\lambda = \lambda y + (1-\lambda)\, x$. Let

$$\lambda^{\max} := \sup\left\{\lambda \in [0,1] \,\middle|\, \left\|\log \zeta_{u_\lambda} - \log \zeta_x\right\|_\infty \leq 16t\right\}. \tag{B.69}$$

It suffices to establish that $\lambda^{\max} = 1$. Note that $\lambda = 0$ is feasible on the RHS of equation (B.69) and hence $\lambda^{\max}$ exists. Now for any $\lambda \in [0, \lambda^{\max}]$ and $i \in \{1, \ldots, n\}$, there exists v on the segment $\overline{u_\lambda x}$ such that

$$\begin{aligned}
&\left|\log \zeta_{u_\lambda, i} - \log \zeta_{x,i}\right| \\
&= \left|\left(\frac{\nabla \zeta_{v,i}}{\zeta_{v,i}}\right)^\top (u_\lambda - x)\right| \\
&\overset{(i)}{\leq} \left\|G_v^{-1} G_v' \,(y-x)\right\|_\infty \\
&= 2\left\|\left(G_v - \alpha \Lambda_v\right)^{-1} \Lambda_v A_v \,(y-x)\right\|_\infty.
\end{aligned}$$

where in step (i) we have used the fact that $u_\lambda - x = \lambda(y-x)$ and $\lambda \in [0,1]$. We claim that

$$\left\|\left(G_v - \alpha \Lambda_v\right)^{-1} \Lambda_v u_1\right\|_\infty \leq \kappa \left\|u_1\right\|_\infty + 2\kappa^2 \left\|G_v^{1/2} u_1\right\|_2 \quad \text{for any } u_1 \in \mathbb{R}^n. \tag{B.70}$$

We prove the claim at the end of this section. We now derive bounds for the two terms on the RHS of the equation (B.70) for $u_1 = A_v(y-x)$. Note that

$$\left\|A_v \,(y-x)\right\|_\infty = \max_i \left|\frac{s_{y,i} - s_{x,i}}{s_{v,i}}\right| = \max_i \left|\frac{s_{y,i} - s_{x,i}}{s_{x,i}}\right| \left|\frac{s_{x,i}}{s_{v,i}}\right| \overset{(i)}{\leq} \frac{2t}{\kappa^2\left(1 - 2t/\kappa^2\right)} \overset{(ii)}{\leq} \frac{3t}{\kappa^2}.$$

Inequality (i) uses bound (B.66) and inequality (ii) follows by plugging in $t \leq 1/64$. Next, we have

$$\begin{aligned}\left\|G_v^{1/2} A_v (y-x)\right\|_2^2 &= \sum_{i=1}^n \zeta_{x,i} \frac{\left(a_i^\top (y-x)\right)^2}{s_{x,i}^2} \frac{\zeta_{v,i}}{\zeta_{x,i}} \frac{s_{v,i}^2}{s_{x,i}^2} \\ &\overset{(i)}{\leq} \|x-y\|_x^2 \max_{i\in[n]} \frac{\zeta_{v,i}}{\zeta_{x,i}} \frac{s_{v,i}^2}{s_{x,i}^2} \\ &\overset{(ii)}{\leq} \frac{t^2}{\kappa^4}\left(1+(16t)+(16t)^2\right)\left(1+\frac{2t}{\kappa^2}\right)^2 \\ &\overset{(iii)}{\leq} \frac{1.5t}{\kappa^4},\end{aligned}$$

where step (i) follows from the definition of the local norm; step (ii) follows from bounds (B.66) and (B.69) and the fact that $e^x \leq 1+x+x^2$ for all $x \in [0, 1/4]$; and inequality (iii) follows by plugging in $t \leq 1/64$. Putting the pieces together, we obtain

$$\|\log \zeta_{u_\lambda} - \log \zeta_x\|_\infty \leq 2(\kappa \cdot 3t/\kappa^2 + 2\kappa^2 \cdot 1.5t/\kappa^4) \leq 12t < 16t.$$

The strict inequality is valid for $\lambda = \lambda^{\max}$. Consequently, using the continuity of $x \mapsto \log \zeta_x$, we conclude that $\lambda^{\max} = 1$.

It is left to prove claim (B.70). Let $v := (G_v - \alpha\Lambda_v)^{-1} \Lambda_v u_1$. which implies $(G_v - \alpha\Lambda_v) v = \Lambda_v u_1$. Plugging the expression of G_v and Λ_v, we have

$$\left((1-\alpha)\Sigma_v + \beta_{\mathrm{J}} \mathbb{I}_n + \alpha \Upsilon_v^{(2)}\right) v = \left(\Sigma_v - \Upsilon_v^{(2)}\right) u_1.$$

Writing component wise, we find that for any $i \in [n]$, we have

$$\begin{aligned}\left|\left((1-\alpha)\sigma_{v,i} + \beta_{\mathrm{J}}\right) v_i\right| &\leq \alpha \left|e_i^\top \Upsilon_v^{(2)} v\right| + \sigma_{v,i} |u_{1,i}| + \left|e_i^\top \Upsilon_v^{(2)} u_1\right| \\ &\overset{(i)}{\leq} \alpha\sigma_{v,i} \left\|\Sigma_v^{1/2} v\right\|_2 + \sigma_{v,i} \|u_1\|_\infty + \sigma_{v,i} \left\|\Sigma_v^{1/2} u_1\right\|_2 \\ &\overset{(ii)}{\leq} \alpha\sigma_{v,i} \left\|G_v^{1/2} v\right\|_2 + \sigma_{v,i} \|u_1\|_\infty + \sigma_{v,i} \left\|G_v^{1/2} u_1\right\|_2 \\ &\overset{(iii)}{\leq} \alpha\sigma_{v,i}\kappa \left\|G_v^{1/2} u_1\right\|_2 + \sigma_{v,i} \|u_1\|_\infty + \sigma_{v,i} \left\|G_v^{1/2} u_1\right\|_2, \qquad \text{(B.71)}\end{aligned}$$

where inequality (ii) from the fact that $\Sigma_y \preceq G_y$ and inequality (iii) from Lemma 35 with $c_1 = 0, c_2 = 1$. To assert inequality (i), observe the following

$$\left|\sum_{j=1}^n \sigma_{y,i,j}^2 v_j\right| \leq \sum_{j=1}^n \sigma_{y,i,j}^2 |v_j| \overset{(a)}{\leq} \sigma_{y,i} \sum_{j=1}^n \sigma_{y,j} |v_j| \overset{(b)}{\leq} \sigma_{y,i} \sum_{j=1}^n \sqrt{\sigma_{y,j}} |v_j| = \sigma_{y,i} \left\|\Sigma_v^{1/2} v\right\|_2,$$

where step (a) follows from the fact that $\sigma_{y,i,j}^2 \leq \sigma_{y,i}\sigma_{y,j}$, and step (b) from the fac that $\sigma_{y,i} \in [0,1]$. Dividing both sides of inequality (B.71) by $((1-\alpha)\sigma_{v,i} + \beta_{\mathrm{J}})$ and observing that $\sigma_{v,i} / ((1-\alpha)\sigma_{v,i} + \beta_{\mathrm{J}}) \leq \kappa$, and $\alpha \in [0,1]$, yields the claim.

B.7 Proof of Lemma 31

We prove Lemma 31 in two parts: claim (B.55a) in Section B.7.1 and claim (B.55b) in Section B.7.2.

B.7.1 Proof of claim (B.55a)

Using the second order Taylor expansion, we have

$$\Psi_z - \Psi_x = (z-x)^\top \nabla\Psi_x + \frac{1}{2}(z-x)^\top \nabla^2\Psi_y (z-x), \quad \text{for some } y \in \overline{xz}.$$

We claim that for $r \le h(\epsilon)$, we have

$$\mathbb{P}\left[(z-x)^\top \nabla\Psi_x \ge -\epsilon/2\right] \ge 1-\epsilon/2, \text{ and} \tag{B.72a}$$

$$\mathbb{P}\left[\frac{1}{2}(z-x)\nabla^2\Psi_y (z-x) \ge -\epsilon/2\right] \ge 1-\epsilon/2. \tag{B.72b}$$

Note that the claim (B.55a) follows from the above two claims.

Proof of bound (B.72a)

We observe that

$$(z-x)^\top \nabla\Psi_x \sim \mathcal{N}\left(0, \frac{r^2}{\kappa^2 n}\nabla\Psi_x^\top J_x^{-1}\nabla\Psi_x\right).$$

Let $E_x = \mathbb{I}_n + (G_x - \alpha\Lambda_x)^{-1}\Lambda_x$. Substituting the expression of $\nabla\Psi_x$ from Lemma 34 (c) and applying Cauchy-Schwarz inequality, we have that for any vector $u \in \mathbb{R}^d$

$$u^\top \nabla\Psi_x \nabla\Psi_x^\top u = (\theta_x^\top G_x E_x A_x u)^2 \le \left(u^\top A_x^\top G_x A_x u\right)\cdot\left(\theta_x^\top G_x E_x G_x^{-1} E_x G_x \theta_x\right). \tag{B.73}$$

Observe that

$$G_x^{1/2} E_x G_x^{-1/2} = \mathbb{I}_n + (\mathbb{I}_n - \alpha G_x^{-1/2}\Lambda_x G_x^{-1/2})^{-1}(G_x^{-1/2}\Lambda_x G_x^{-1/2}).$$

Now, using the intermediate bound (B.100) from the proof of Lemma 35, we obtain that

$$\mathbb{I}_n \preceq G_x^{1/2} E_x G_x^{-1/2} \preceq 2\kappa \mathbb{I}_n,$$

and hence $G_x \preceq G_x E_x G_x^{-1} E_x G_x \preceq 4\kappa^2 G_x$. Consequently, we have

$$\theta_x^\top G_x E_x G_x^{-1} E_x G_x \theta_x \le 4\kappa^2 \theta_x^\top G_x \theta_x = 4\kappa^2 \sum_{i=1}^n \zeta_{x,i}\theta_{x,i}^2 \le 16\kappa^2 d,$$

where the last step follows from Lemma 32. Putting the pieces together into equation (B.73), we obtain $\nabla\Psi_x\nabla\Psi_x^\top \preceq 16\kappa^2 d J_x$ whence $J_x^{-1/2}\nabla\Psi_x\nabla\Psi_x^\top J_x^{-1/2} \preceq 16\kappa^2 d\mathbb{I}_d$. Noting that the matrix $J_x^{-1/2}\nabla\Psi_x\nabla\Psi_x^\top J_x^{-1/2}$ has rank one, we have

$$\nabla\Psi_x^\top J_x^{-1}\nabla\Psi_x = \text{trace}\left(J_x^{-1/2}\nabla\Psi_x\nabla\Psi_x^\top J_x^{-1/2}\right) \leq 16\kappa^2 d.$$

Using standard Gaussian tail bound, we have $\mathbb{P}\left((z-x)^\top \nabla\Psi_x \geq -\sqrt{32}\chi_1 r\right) \geq 1 - \exp\left(-\chi_1^2\right)$.

Choosing $\chi_1 = \log(2/\epsilon)$, and observing that

$$r \leq \frac{\epsilon}{\left(2\sqrt{32}\chi_1\right)}, \tag{B.74}$$

yields the claim.

Proof of bound (B.72b)

In the following proof, we use $h = z - x$ for definitions (B.58a)-(B.58d). According to Lemma 34(e), we have

$$\left|\frac{1}{2}(z-x)^\top \nabla^2\Psi_y (z-x)\right| \leq \sum_{i=1}^n \zeta_{y,i}\,\theta_{y,i}\left[\frac{9}{2}d_{y,i}^2 + 2f_{y,i}^2\right] + \frac{1}{2}\left|\sum_{i=1}^n \zeta_{y,i}\,\theta_{y,i}\ell_{y,i}\right|$$

We claim that

$$\sum_{i=1}^n \zeta_{y,i}\,\theta_{y,i}\left[\frac{9}{2}d_{y,i}^2 + 2f_{y,i}^2\right] + \frac{1}{2}\left|\sum_{i=1}^n \zeta_{y,i}\,\theta_{y,i}\ell_{y,i}\right| \leq 386\sqrt{d}\kappa^4 \sum_{i=1}^n \zeta_{y,i}d_{y,i}^2. \tag{B.75}$$

Assuming the claim as given at the moment, we now complete the proof. Note that y is some particular point on $\overline{xz}$ and its dependence on z is hard to characterize. Consequently, we transfer all the terms with dependence on y, to terms with dependence on x only. We have

$$\sum_{i=1}^n \zeta_{y,i}d_{y,i}^2 = \sum_{i=1}^n \zeta_{x,i}d_{x,i}^2 \underbrace{\frac{\zeta_{y,i}}{\zeta_{x,i}}\frac{s_{x,i}^2}{s_{y,i}^2}}_{\tau_{y,i}}.$$

Using the following high probability bounds implied by Lemma 37 and Lemma 38 (B.65a), we obtain

$$\mathbb{P}\left[\sup_{y\in\overline{xz},i\in[n]} \tau_{y,i} \leq 1.1\right] \geq 1-\epsilon/4, \quad \text{and,} \quad \mathbb{P}\left[\sum_{i=1}^n \zeta_{x,i}\left(\hat{a}_{x,i}^\top\xi\right)^2 \leq \chi_2\sqrt{24}d\right] \geq 1-\epsilon/16. \tag{B.76}$$

Since $h = z - x$, we have that $d_{x,i}^2 = \frac{r^2}{\kappa^2 d^{3/2}} \left(\hat{a}_{x,i}^\top \xi\right)^2$. Consequently, for

$$r \leq \sqrt{\frac{\epsilon}{386\sqrt{24}\chi_2}}, \tag{B.77}$$

with probability at least $1 - \epsilon/2$, we have

$$\left|\frac{1}{2}(z-x)^\top \nabla^2 \Psi_y (z-x)\right| \overset{\text{eqn. (B.75)}}{\leq} 386\sqrt{d}\kappa^4 \sum_{i=1}^n \zeta_{y,i} d_{y,i}^2 \overset{\text{hpb (B.76)}}{\leq} \epsilon,$$

which completes the proof.

We now turn to the proof of claim (B.75). First we observe the following relationship between the terms $d_{y,i}$ and $f_{y,i}$:

$$\sum_{i=1}^n \zeta_{y,i} f_{y,i}^2 \tag{B.78}$$

$$\overset{(i)}{=} 4h^\top A_y^\top \Lambda_y (G_y - \alpha\Lambda_y)^{-1} G_y (G_y - \alpha\Lambda_y)^{-1} \Lambda_y A_y h \tag{B.79}$$

$$\overset{(ii)}{\leq} 4\kappa^2 h^\top A_y^\top G_y A_y h \tag{B.80}$$

$$= 4\kappa^2 \sum_{i=1}^n \zeta_{y,i} d_{y,i}^2, \tag{B.81}$$

where step (i) follows by plugging in the definition of $f_{y,i}$ (B.58b) and step (ii) by invoking Lemma 35 with $c_1 = 0$ and $c_2 = 1$. Next, we relate the term on the LHS of equation (B.75) involving $\ell_{y,i}$ to a polynomial in $d_{y,i}$. Using Lemma 34, we find that

$$\begin{aligned}
&\left|\sum_{i=1}^n \zeta_{y,i}\, \theta_{y,i} \ell_{y,i}\right| \\
&= \left|\left((G_y - \alpha\Lambda_y)^{-1} G_y \theta_y\right)^\top (G_y - \alpha\Lambda_y)\, \ell_y\right| \\
&\leq \left\|\underbrace{(G_y - \alpha\Lambda_y)^{-1} G_y \theta_y}_{v_1}\right\|_\infty \left\|\underbrace{(G_y - \alpha\Lambda_y)\, \ell_y}_{\rho_y}\right\|_1,
\end{aligned}$$

where the last step follows from the Holder's inequality: for any two vectors $u, v \in \mathbb{R}^d$, we have that $u^\top v \leq \|u\|_\infty \|v\|_1$. Substituting the bound for the norm $\|v_1\|_\infty$ from

Corollary 8 and the bound on $\rho_{y,i}$ from Lemma 34(b), we obtain that

$$\begin{aligned}
&\left|\sum_{i=1}^{n} \zeta_{y,i}\, \theta_{y,i} \ell_{y,i}\right| \\
&\leq 12\sqrt{n}\kappa^{3/2} \sum_{i=1}^{n} \left[7\zeta_{y,i} d_{y,i}^2 + 3\zeta_{y,i} f_{y,i}^2 + \sum_{j=1}^{n} \left(13 d_{y,j}^2 + 6 f_{y,j}^2\right) \Upsilon_{y,i,j}^2\right] \\
&\leq 672\sqrt{n}\kappa^4 \sum_{i=1}^{n} \zeta_{y,i} d_{y,i}^2,
\end{aligned}$$

where the last step follows from Lemma 32(a) and the bound (B.81). The claim now follows.

B.7.2 Proof of claim (B.55b)

Writing $z = x + tu$, where t is a scalar and u is a unit vector in $\mathbb{R}^d$, we obtain

$$\|z - x\|_z^2 - \|z - x\|_x^2 = t^2 \sum_{i=1}^{n} \left(a_i^\top u\right)^2 (\varphi_{z,i} - \varphi_{x,i}).$$

Now, we use a Taylor series expansion for $\sum_{i=1}^{n} \left(a_i^\top u\right)^2 (\varphi_{z,i} - \varphi_{x,i})$ around the point x, along the line u. There exists a point $y \in \overline{xz}$ such that

$$\sum_{i=1}^{n} \left(a_i^\top u\right)^2 (\varphi_{z,i} - \varphi_{x,i}) = \sum_{i=1}^{n} \left(a_i^\top u\right)^2 \left((z-x)^\top \nabla\varphi_{x,i} + \frac{1}{2}(z-x)^\top \nabla^2\varphi_{y,i}\,(z-x)\right).$$

Note that the point y in this discussion is not the same as the point y used in previous proofs, in particular in Section B.7.1. Multiplying both sides by t^2, and using the shorthand $d_{x,i} = \frac{a_i^\top (z-x)}{s_{x,i}}$, we obtain

$$\|z-x\|_z^2 - \|z-x\|_x^2 = \sum_{i=1}^{n} d_{x,i}^2 s_{x,i}^2 (z-x)^\top \nabla\varphi_{x,i} + \sum_{i=1}^{n} d_{x,i}^2 s_{x,i}^2 \frac{1}{2}(z-x)^\top \nabla^2\varphi_{y,i}\,(z-x). \tag{B.82}$$

We claim that for $r \leq h(\epsilon)$, we have

$$\mathbb{P}_{z\sim\mathcal{T}_x^{\mathrm{J}}}\left[\sum_{i=1}^{n} d_{x,i}^2 s_{x,i}^2 (z-x)^\top \nabla\varphi_{x,i} \leq \epsilon \frac{r^2}{\kappa^4 d^{3/2}}\right] \geq 1 - \epsilon/2, \text{ and} \tag{B.83a}$$

$$\mathbb{P}_{z\sim\mathcal{T}_x^{\mathrm{J}}}\left[\sup_{y\in\overline{xz}} \left(\sum_{i=1}^{n} d_{x,i}^2 s_{x,i}^2 \frac{1}{2}(z-x)^\top \nabla^2\varphi_{y,i}\,(z-x)\right) \leq \epsilon \frac{r^2}{\kappa^4 d^{3/2}}\right] \geq 1 - \epsilon/2. \tag{B.83b}$$

We now prove each claim separately.

Proof of bound (B.83a)

Using Lemma 34(d) and using $h = z - x$ where z is given by the relation (B.52), we find that

$$\begin{aligned}\sum_{i=1}^{n} d_{x,i}^2 s_{x,i}^2 (z-x)^\top \nabla\varphi_{x,i} &= \sum_{i=1}^{n} \zeta_{x,i} d_{x,i}^2 (2d_{x,i} + f_{x,i}) \\ &= \frac{r^3}{d^{9/4}\kappa^6} \sum_{i=1}^{n} \zeta_{x,i} \left(\hat{a}_{x,i}^\top \xi\right)^3 + \frac{2r^3}{d^{9/4}\kappa^6} \sum_{i=1}^{n} \zeta_{x,i} \left(\hat{a}_{x,i}^\top \xi\right)^2 \left(\hat{b}_{x,i}^\top \xi\right)\end{aligned} \tag{B.84}$$

Using high probability bounds for the two terms in equation (B.84) from Lemma 38, part (B.65b) and part (B.65c), we obtain that

$$\left|\sum_{i=1}^{n} d_{x,i}^2 s_{x,i}^2 (z-x)^\top \nabla\varphi_{x,i}\right| \leq \frac{5\sqrt{60}\chi_3 r^3}{\kappa^5 d^{7/4}} \leq \epsilon \frac{r^2}{\kappa^4 d^{3/2}},$$

with probability at least $1 - \epsilon/2$. The last inequality uses the condition that

$$r \leq \frac{\epsilon}{5\sqrt{60}\chi_3}. \tag{B.85}$$

The claim now follows.

Proof of bound (B.83b)

Note that $d_{x,i} s_{x,i} = a_i^\top h = d_{y,i} s_{y,i}$ for any h. Using this equality for $h = z - x$, we find that

$$\begin{aligned}\left|\sum_{i=1}^{n} d_{x,i}^2 s_{x,i}^2 \frac{1}{2} h^\top \nabla^2 \varphi_{y,i} h\right| &= \left|\sum_{i=1}^{n} d_{y,i}^2 s_{y,i}^2 \frac{1}{2} h^\top \nabla^2 \varphi_{y,i} h\right| \\ &\overset{(i)}{\leq} 3 \underbrace{\sum_{i=1}^{n} \zeta_{y,i} d_{y,i}^4}_{C_1} + 2 \underbrace{\left|\sum_{i=1}^{n} \zeta_{y,i} d_{y,i}^3 f_{y,i}\right|}_{C_2} + \underbrace{\left|\sum_{i=1}^{n} \zeta_{y,i} d_{y,i}^2 \ell_{y,i}\right|}_{C_3},\end{aligned} \tag{B.86}$$

where step (i) follows from Lemma 34(f). We can write C_1 as follows

$$\sum_{i=1}^{n} \zeta_{y,i} d_{y,i}^4 = \sum_{i=1}^{n} \zeta_{x,i} d_{x,i}^4 \frac{\zeta_{y,i}}{\zeta_{x,i}} \frac{d_{y,i}^4}{d_{x,i}^4} = \frac{r^4}{n^3\kappa^8} \sum_{i=1}^{n} \zeta_{x,i} \left(\hat{a}_{x,i}^\top \xi\right)^4 \frac{\zeta_{y,i}}{\zeta_{x,i}} \frac{d_{y,i}^4}{d_{x,i}^4}. \tag{B.87}$$

Now, we claim the following:

$$C_2 \leq 2\frac{r^4}{n^3\kappa^7} \cdot \sqrt{\left[\sum_{i=1}^{n} \zeta_{x,i} \left(\hat{a}_{x,i}^\top \xi\right)^2 \frac{\zeta_{y,i}}{\zeta_{x,i}} \frac{d_{y,i}^2}{d_{x,i}^2}\right] \cdot \left[\sum_{i=1}^{n} \zeta_{x,i} \left(\hat{a}_{x,i}^\top \xi\right)^6 \frac{\zeta_{y,i}}{\zeta_{x,i}} \frac{d_{y,i}^6}{d_{x,i}^6}\right]}, \quad \text{and,} \tag{B.88a}$$

$$C_3 \leq 56\frac{r^4}{n^3\kappa^{4.5}} \left(\sum_{i=1}^{n} \zeta_{x,i} \left(\hat{a}_{x,i}^\top \xi\right)^2 \frac{\zeta_{y,i}}{\zeta_{x,i}} \frac{d_{y,i}^2}{d_{x,i}^2}\right) \left(\max_i \left(\hat{a}_{x,i}^\top \xi\right)^2 \frac{d_{y,i}^2}{d_{x,i}^2} + \sqrt{\sum_{i=1}^{n} \zeta_{x,i} \left(\hat{a}_{x,i}^\top \xi\right)^4 \frac{\zeta_{y,i}}{\zeta_{x,i}} \frac{d_{y,i}^4}{d_{x,i}^4}}\right) \tag{B.88b}$$

Assuming the claims as given, we now complete the proof. Using Lemma 37, we have

$$\mathbb{P}\left[\frac{\zeta_{y,i}}{\zeta_{x,i}}\frac{d_{y,i}^6}{d_{x,i}^6} \leq 1.2\right] \geq 1 - \epsilon/4,$$

and consequently

$$3C_1+2C_2+C_3 \leq \frac{r^4}{d^3\kappa^{4.5}}\Bigg[4\cdot\sum_{i=1}^n \zeta_{x,i}(\hat{a}_{x,i}^\top\xi)^4 + 10\cdot\Bigg(\sum_{i=1}^n \zeta_{x,i}(\hat{a}_{x,i}^\top\xi)^2\cdot\sum_{i=1}^n \zeta_{x,i}(\hat{a}_{x,i}^\top\xi)^6\Bigg)^{1/2} + 100\cdot\sum_{i=1}^n \zeta_{x,i}\left(\hat{a}_{x,i}^\top\xi\right)^2\cdot\left(\max_i(\hat{a}_{x,i}^\top\xi)^2 + \Big(\sum_{i=1}^n \zeta_{x,i}(\hat{a}_{x,i}^\top\xi)^4\Big)^{1/2}\right)\Bigg], \tag{B.89}$$

with probability at least $1-\epsilon/4$. Now, we observe that for all $i \in [n]$ and $x \in \operatorname{int}(\mathcal{K})$, we have

$$\left(\hat{a}_{x,i}^\top\xi\right) \sim \mathcal{N}(0,\theta_{x,i}) \quad \text{and} \quad \theta_{x,i} \leq 4.$$

Invoking the standard tail bound for maximum of Gaussian random variables, we obtain

$$\mathbb{P}\left[\max_i\left|\left(\hat{a}_{x,i}^\top\xi\right)\right| \leq 8\cdot\left(\sqrt{\log n} + \sqrt{\log(32/\epsilon)}\right)\right] \geq 1-\epsilon/16.$$

Using the fact that $2c_1c_2 \geq c_1+c_2$ for all $c_1,c_2 \geq 1$, we obtain

$$\mathbb{P}\left[\max_i\left|\left(\hat{a}_{x,i}^\top\xi\right)\right| \leq 16\cdot\sqrt{\log n}\cdot\sqrt{\log(32/\epsilon)}\right] \geq 1-\epsilon/16.$$

Combining this bound with the tail bounds for various Gaussian polynomials (B.65a), (B.65d), (B.65e) from Lemma 38, and substituting in inequality (B.89), we obtain that

$$\left|\sum_{i=1}^n d_{x,i}^2 s_{x,i}^2 \frac{1}{2}h^\top\nabla^2\varphi_{y,i}h\right| \leq \frac{r^4}{\kappa^{6.5}d^3}\Bigg[4\cdot\chi_4\sqrt{1680}d + 10\left(\chi_2\sqrt{24}d\cdot\chi_6\sqrt{15120}d\right)^{1/2} + 100\cdot\chi_2\sqrt{24}d\cdot\left(256\cdot\log n\cdot\log(32/\epsilon) + \left(\chi_4\sqrt{1680}d\right)^{1/2}\right)\Bigg]$$

with probability at least $1-\epsilon/2$. In the above expression, the terms χ_i are a function of ϵ as defined in Lemma 38. In particular, $\chi_i = \chi_{i,\epsilon} = (2e/i\cdot\log(16/\epsilon))^{i/2}$ for $i \in \{2,3,4,6\}$. Observing that $256\log(32/\epsilon) \geq \left(\chi_4\sqrt{1680}\right)^{1/2}$, and that our choice of r satisfies

$$r^2 \leq \min\left\{\frac{\epsilon}{8\sqrt{1680}\chi_4}, \frac{\epsilon}{40\left(\chi_2\chi_6\sqrt{24}\sqrt{15120}\right)^{1/2}}, \frac{\epsilon}{204800\chi_2\sqrt{24}\log(32/\epsilon)}\right\}, \tag{B.90}$$

we obtain

$$\left|\sum_{i=1}^{n} d_{x,i}^2 s_{x,i}^2 \frac{1}{2} h^\top \nabla^2 \varphi_{y,i} h\right| \leq \frac{r^2}{\kappa^4 d^{3/2}} \left[\frac{\epsilon}{2} + \frac{\epsilon}{4} + \frac{\epsilon}{8}\left(\frac{\log n}{\sqrt{d}} + 1\right)\right].$$

Asserting the additional condition $\sqrt{d} \geq \log n$, yields the claim.

It is now left to prove the bounds (B.88a) and (B.88b). We prove these bounds separately.

Bounding C_2: Applying Cauchy-Schwarz inequality, we have

$$\left|\sum_{i=1}^{n} \zeta_{y,i} d_{y,i}^3 f_{y,i}\right| \leq \left(\sum_{i=1}^{n} \zeta_{y,i} f_{y,i}^2 \cdot \sum_{i=1}^{n} \zeta_{y,i} d_{y,i}^6\right)^{1/2}$$

Using the bound (B.81), we obtain

$$\sum_{i=1}^{n} \zeta_{y,i} f_{y,i}^2 \leq 4\kappa^2 \sum_{i=1}^{n} \zeta_{y,i} d_{y,i}^2 = 4\kappa^2 \sum_{i=1}^{n} \zeta_{x,i} d_{x,i}^2 \frac{\zeta_{y,i}}{\zeta_{x,i}} \frac{d_{y,i}^2}{d_{x,i}^2}.$$

Substituting $h = z - x$ where z is given by relation (B.52), we obtain that $d_{x,i} = \frac{r}{d^{3/4}\kappa} \hat{a}_{x,i}^\top \xi$, and thereby

$$\sum_{i=1}^{n} \zeta_{y,i} f_{y,i}^2 \leq 4\kappa^2 \frac{r^2}{d^{3/2}\kappa^4} \sum_{i=1}^{n} \zeta_{x,i} (\hat{a}_{x,i}^\top \xi)^2 \frac{\zeta_{y,i}}{\zeta_{x,i}} \frac{d_{y,i}^2}{d_{x,i}^2}.$$

Doing similar algebra, we obtain $\sum_{i=1}^{n} \zeta_{y,i} d_{y,i}^6 = \frac{r^6}{d^{9/2}\kappa^{12}} \sum_{i=1}^{n} \zeta_{x,i} \left(\hat{a}_{x,i}^\top \xi\right)^6 \frac{\zeta_{y,i}}{\zeta_{x,i}} \frac{d_{y,i}^6}{d_{x,i}^6}$. Putting the pieces together yields the claim.

Bounding C_3: Recall that $\rho_y = (G_y - \alpha\Lambda_y)\ell_y$ (Lemma 34) and $\mu_y = (G_y - \alpha\Lambda_y)^{-1} G_y$ (Lemma 36). We have

$$\left|\sum_{i=1}^{n} \zeta_{y,i} d_{y,i}^2 \ell_{y,i}\right| = \mathbf{1} D_y^2 G_y \ell_y = \underbrace{\mathbf{1} D_y^2 G_y (G_y - \alpha\Lambda_y)^{-1}}_{=:u_y^\top} \underbrace{(G_y - \alpha\Lambda_y)\ell_y}_{\rho_y}.$$

Using the definition of u_y and μ_y, we obtain

$$u_{y,i} := e_i^\top u_y = e_i^\top (G_y - \alpha\Lambda_y)^{-1} G_y D_y^2 \mathbf{1} = e_i^\top \mu_y D_y^2 \mathbf{1} = \mu_{y,i,i} d_{y,i}^2 + \sum_{j \in [n], j \neq i} \mu_{y,i,j} d_{y,j}^2.$$

Consequently, we have

$$\left|\sum_{i=1}^{n} u_{y,i} \rho_{y,i}\right| \leq \overbrace{\sum_{i=1}^{n} |\rho_{y,i}| \cdot \left|\mu_{y,i,i} d_{y,i}^2\right|}^{=:C_4} + \overbrace{\sum_{i=1}^{n} |\rho_{y,i}| \cdot \left(\sum_{j \in [n], j \neq i} \left|\mu_{y,i,j} d_{y,j}^2\right|\right)}^{=:C_5}$$

From Lemma 36, we have that $\mu_{y,i,i} \in [0, \kappa]$. Hence, we have $C_4 \leq \|\rho_y\|_1 \cdot \kappa \cdot \max_{i\in[n]} d_{y,i}^2$. To bound C_5, we note that

$$\sum_{j\in[n],j\neq i} \left|\mu_{y,i,j} d_{y,j}^2\right| \overset{(i)}{\leq} \left(\sum_{j\in[n],j\neq i} \frac{\mu_{y,i,j}^2}{\zeta_{y,j}} \cdot \sum_{j=1}^n \zeta_{y,j} d_{y,j}^4\right)^{1/2} \overset{(ii)}{\leq} \left(\kappa^3 \cdot \sum_{j=1}^n \zeta_{x,j} d_{x,j}^4 \frac{\zeta_{y,j}}{\zeta_{x,j}} \frac{d_{y,j}^4}{d_{x,j}^4}\right)^{1/2},$$

where step (i) follows from Cauchy-Schwarz inequality and step (ii) from Lemma 36. Putting the pieces together, we obtain that

$$\left|\sum_{i=1}^n \zeta_{y,i} d_{y,i}^2 \ell_{y,i}\right| \leq \|\rho_y\|_1 \cdot \left[\kappa \cdot \max_{i\in[n]} d_{y,i}^2 + \kappa^{3/2} \left(\sum_{j=1}^n \zeta_{x,j} d_{x,j}^4 \frac{\zeta_{y,j}}{\zeta_{x,j}} \frac{d_{y,j}^4}{d_{x,j}^4}\right)^{1/2}\right].$$

Using the bound on $\|\rho_y\|_1$ from Lemma 34, we have

$$\left|\sum_{i=1}^n \zeta_{y,i} d_{y,i}^2 \ell_{y,i}\right| \leq \left(56\kappa^2 \sum_{i=1}^n \zeta_{y,i} d_{y,i}^2\right) \cdot \left[\kappa \cdot \max_{i\in[n]} d_{y,i}^2 + \kappa^{3/2} \left(\sum_{j=1}^n \zeta_{x,j} d_{x,j}^4 \frac{\zeta_{y,j}}{\zeta_{x,j}} \frac{d_{y,j}^4}{d_{x,j}^4}\right)^{1/2}\right].$$

Substituting the expression for $d_{x,i} = \frac{r}{\kappa^2 d^{3/4}} \left(\hat{a}_{x,i}^\top \xi\right)$ yields the claim.

B.8 Proofs of Lemmas from Section B.5.1

In this section we collect proofs of lemmas from Section B.5.1. Each lemma is proved in a different subsection.

B.8.1 Proof of Lemma 34

Up to second order terms, we have

$$\frac{1}{s_{x+h,i}^2} = \frac{1}{s_{x,i}^2}\left[1 + \frac{2a_i^\top h}{s_{x,i}} + \frac{3(a_i^\top h)^2}{s_{x,i}^2}\right] + O\left(\|h\|_2^3\right), \tag{B.91a}$$

$$\zeta_{y+h,i} = \zeta_{y,i} + h^\top \nabla \zeta_{y,i} + \frac{1}{2} h^\top \nabla^2 \zeta_{y,i} h + O\left(\|h\|_2^3\right), \tag{B.91b}$$

$$\zeta_{y+h,i}^\alpha = \zeta_{y,i}^\alpha + \alpha \zeta_{y,i}^{\alpha-1}\left(h^\top \nabla \zeta_{y,i} + \frac{1}{2} h^\top \nabla^2 \zeta_{y,i} h\right) + \frac{\alpha(\alpha-1)}{2} \zeta_{y,i}^{\alpha-2}\left(h^\top \nabla \zeta_{y,i}\right)^2 + O\left(\|h\|_2^3\right), \tag{B.91c}$$

Further, let

$$\tilde{J}_y := A_y^\top G_y^\alpha A_y = \sum_{i=1}^n \zeta_{y,i}^\alpha \frac{a_i a_i^\top}{s_{y,i}^2}. \tag{B.91d}$$

Using equations (B.91a) and (B.91c), and substituting $d_{y,i} = a_i^\top h / s_{y,i}$, $f_{y,i} = h^\top \nabla \zeta_{y,i} / \zeta_{y,i}$ and $\ell_{y,i} = \frac{1}{2} h^\top \nabla^2 \zeta_{y,i} h / \zeta_{y,i}$, we find that

$$\tilde{J}_{y+h} = \sum_{i=1}^{n} \left[1 + \alpha f_{y,i} + \alpha \ell_{y,i} + \frac{\alpha(\alpha-1)}{2} f_{y,i}^2\right] \left[1 + 2d_{y,i} + 3d_{y,i}^2\right] \zeta_{y,i}^\alpha \frac{a_i a_i^\top}{s_{y,i}^2} + O\left(\|h\|_2^3\right).$$

Note that $d_{y,i}$ and $f_{y,i}$ are first order terms in $\|h\|_2$ and $\ell_{y,i}$ is a second order term in $\|h\|_2$.

Thus we obtain

$$\begin{aligned}\tilde{J}_{y+h} - \tilde{J}_y = &\underbrace{\sum_{i=1}^{n} \left(2d_{y,i} + \alpha f_{y,i}\right) \zeta_{y,i}^\alpha \frac{a_i a_i^\top}{s_{y,i}^2}}_{=:\Delta_{y,h}^{(1)}} \\ &+ \underbrace{\sum_{i=1}^{n} \left[3d_{y,i}^2 + 2\alpha d_{y,i} f_{y,i} + \alpha \ell_{y,i} + \frac{\alpha(\alpha-1)}{2} f_{y,i}^2\right] \zeta_{y,i}^\alpha \frac{a_i a_i^\top}{s_{y,i}^2}}_{=:\Delta_{y,h}^{(2)}} + O\left(\|h\|_2^3\right).\end{aligned}$$

Let $\Delta_{y,h} := \Delta_{y,h}^{(1)} + \Delta_{y,h}^{(2)}$. Note that $\Delta_{y,h}^{(i)}$ denotes the i-th order term in $\|h\|_2$. Finally, the following expansion also comes in handy for our derivations:

$$a_i^T \tilde{J}_{y+h}^{-1} a_i = a_i^\top \tilde{J}_y^{-1} a_i - a_i^\top \tilde{J}_y^{-1} \Delta_{y,h} \tilde{J}_y^{-1} a_i + a_i^\top \tilde{J}_y^{-1} \Delta_{y,h} \tilde{J}_y^{-1} \Delta_{y,h} \tilde{J}_y^{-1} a_i + O\left(\|h\|_2^3\right). \tag{B.91e}$$

Proof of part (a): Gradient of weights

The expression for the gradient $\nabla \zeta_{y,i}$ is derived in Lemma 14 of the paper [107] and is thereby omitted.

Proof of part (b): Hessian of weights

We claim that

$$\begin{aligned}\rho_y = \left(\mathbb{I} - \alpha \Lambda_y G_y^{-1}\right) \begin{bmatrix} \frac{1}{2} h^\top \nabla^2 \zeta_{y,1} h \\ \cdots \\ \frac{1}{2} h^\top \nabla^2 \zeta_{y,m} h \end{bmatrix} &= (2D_y + \alpha F_y) \Upsilon_y^{(2)} (2D_y + \alpha F_y) \mathbf{1} \\ &+ \left(\Sigma_y - \Upsilon_y^{(2)}\right) \left[2\alpha D_y F_y + 3D_y^2 + \tau_\alpha F_y^2\right] \mathbf{1} \\ &+ \operatorname{diag}\left(\Upsilon_y (2D_y + \alpha F_y) \Upsilon_y (2D_y + \alpha F_y) \Upsilon_y\right),\end{aligned} \tag{B.92}$$

where we have used $\operatorname{diag}(B)$ to denote the diagonal vector $(B_{1,1}, \ldots, B_{n,n})$ of the matrix B. Deferring the proof of this expression for the moment, we now derive a bound on

the ℓ_1 norm of ρ_y. Expanding the i-th term of $\rho_{y,i}$ from equation (B.92), we obtain

$$\rho_{y,i} = (2d_{y,i} + \alpha f_{y,i}) \sum_{j=1}^{n} (2d_{y,j} + \alpha f_{y,j}) \Upsilon_{y,i,j}^2 + \left[2\alpha d_{y,i} f_{y,i} + 3d_{y,i}^2 + \tau_\alpha f_{y,i}^2\right] \sigma_{y,i}$$
$$- \sum_{j=1}^{n} \left[2\alpha d_{y,j} f_{y,j} + 3d_{y,j}^2 + \tau_\alpha f_{y,j}^2\right] \Upsilon_{y,i,j}^2 + \sum_{j,l=1}^{n} (2d_{y,j} + \alpha f_{y,j})(2d_{y,l} + \alpha f_{y,l}) \Upsilon_{y,i,j} \Upsilon_{y,j,l} \Upsilon_{y,l,i}.$$

Recall that $\alpha = 1 - 1/\log_2(2n/d)$. Since $n \geq d$ for polytopes, we have $\alpha \in [0,1]$ and consequently $|\tau_\alpha| = |\alpha(\alpha-1)/2| \in [0,1]$. Further note that Υ_x is an orthogonal projection matrix, and hence we have

$$\operatorname{diag}(\Upsilon_x e_i) \Upsilon_x \operatorname{diag}(\Upsilon_x e_i) \preceq \operatorname{diag}(\Upsilon_x e_i) \operatorname{diag}(\Upsilon_x e_i).$$

Combining these observations with the AM-GM inequality, we have

$$|\rho_{y,i}| \leq 7\sigma_{y,i} d_{y,i}^2 + 3\sigma_{y,i} f_{y,i}^2 + \sum_{j=1}^{n} \left(13 d_{y,j}^2 + 6 f_{y,j}^2\right) \Upsilon_{y,i,j}^2.$$

Summing both sides over the index i, we find that

$$\sum_{i=1}^{n} |\rho_{y,i}| \overset{(i)}{\leq} \sum_{i=1}^{n} 20\sigma_{y,i} d_{y,i}^2 + 9\sigma_{y,i} f_{y,i}^2 \overset{(ii)}{\leq} \sum_{i=1}^{n} 20\zeta_{y,i} d_{y,i}^2 + 9\zeta_{y,i} f_{y,i}^2 \overset{(iii)}{\leq} 56\kappa^2 \sum_{i=1}^{n} \zeta_{y,i} d_{y,i}^2,$$

where step (i) follows from Lemma 32 (a), step (ii) from Lemma 28 (a) and step (iii) from the bound (B.81).

We now return to the proof of expression (B.92). Using equation (B.57c), we find that

$$\frac{1}{2} h^\top \nabla^2 \zeta_{y,i} h = \frac{1}{2} h^\top \nabla^2 \sigma_{y,i} h \quad \text{for all } i \in [n]. \tag{B.93}$$

Next, we derive the Taylor series expansion of $\sigma_{y,i}$. Using the definition of $\tilde{J}_x$ (B.91d) in equation (B.42), we find that $\sigma_{y,i} = \zeta_{y,i}^\alpha \frac{a_i^\top \tilde{J}_y^{-1} a_i}{s_{y,i}^2}$. To compute the difference $\sigma_{y+h,i} - \sigma_{y,i}$, we use the expansions (B.91a), (B.91c) and (B.91e). Letting $\tau_\alpha = \alpha(\alpha-1)/2$, we have

$$\begin{aligned}
\sigma_{y+h,i} &= \zeta_{y+h,i}^\alpha \frac{a_i^\top \tilde{J}_{y+h}^{-1} a_i}{s_{y+h,i}^2} \\
&= \zeta_{y,i}^\alpha \frac{a_i^\top \tilde{J}_{y+h}^{-1} a_i}{s_{y,i}^2} \left[1 + \alpha f_{y,i} + \alpha \ell_{y,i} + \tau_\alpha f_{y,i}^2\right] \left[1 + 2d_{y,i} + 3d_{y,i}^2\right] + O\left(\|h\|_2^3\right) \\
&= \sigma_{y,i} + (2d_{y,i} + \alpha f_{y,i}) \sigma_{y,i} - \sum_{j=1}^{n} (2d_{y,j} + \alpha f_{y,j}) \Upsilon_{y,i,j}^2 + (2d_{y,i} + \alpha f_{y,i}) \sum_{j=1}^{n} (2d_{y,j} + \alpha f_{y,j}) \Upsilon_{y,i,j}^2 \\
&\quad + 2\alpha d_{y,i} f_{y,i} \sigma_{y,i} + \left[\alpha \ell_{y,i} + \tau_\alpha f_{y,i}^2 + 3d_{y,i}^2\right] \sigma_{y,i} - \sum_{j=1}^{n} \left[3d_{y,j}^2 + 2\alpha d_{y,j} f_{y,j} + \alpha \ell_{y,j} + \tau_\alpha f_{y,j}^2\right] \Upsilon_{y,i,j}^2 \\
&\quad + \sum_{j,l=1}^{n} (2d_{y,j} + \alpha f_{y,j})(2d_{y,l} + \alpha f_{y,l}) \Upsilon_{y,i,j} \Upsilon_{y,j,l} \Upsilon_{y,l,i} + O\left(\|h\|_2^3\right).
\end{aligned}$$

We identify the second order (in $O\left(\|h\|_2^2\right)$) terms in the previous expression. Using the equation (B.93), these are indeed the terms that correspond to the terms $\frac{1}{2}h^\top \nabla^2 \zeta_{y,i} h$, $i \in [n]$. Substituting $\ell_{y,i} = \frac{1}{2}h^\top \nabla^2 \zeta_{y,i} h / \zeta_{y,i}$, we have

$$\begin{aligned}
&\frac{1}{2}h^\top \nabla^2 \zeta_{y,i} h \\
&= (2d_{y,i} + \alpha f_{y,i}) \sum_{j=1}^{n} (2d_{y,j} + \alpha f_{y,j}) \Upsilon_{y,i,j}^2 + 2\alpha d_{y,i} f_{y,i} \sigma_{y,i} + \left[\frac{\alpha}{2} \frac{h^\top \nabla^2 \zeta_{y,i} h}{\zeta_{y,i}} + \tau_\alpha f_{y,i}^2 + 3d_{y,i}^2 \right] \sigma_{y,i} \\
&- \sum_{j=1}^{n} \left[3d_{y,j}^2 + 2\alpha d_{y,j} f_{y,j} + \frac{\alpha}{2} \frac{h^\top \nabla^2 \zeta_{y,j} h}{\zeta_{y,j}} + \tau_\alpha f_{y,j}^2 \right] \Upsilon_{y,i,j}^2 + \sum_{j,l=1}^{n} (2d_{y,j} + \alpha f_{y,j})(2d_{y,l} + \alpha f_{y,l}) \Upsilon_{y,i,j} \Upsilon_{y,j,l} \Upsilon_{y,l,i}.
\end{aligned}$$

Collecting the different terms and doing some algebra yields the result (B.92).

Proof of part (c): Gradient of logdet

For a unit vector $h \in \mathbb{R}^d$, we have

$$\begin{aligned}
h^\top \log\det J_y &= \lim_{\delta \to 0} \frac{1}{\delta} (\log\det J_{y+\delta h} - \log\det J_y) \\
&= \lim_{\delta \to 0} \frac{1}{\delta} (\log\det J_y^{-1/2} J_{y+\delta h} J_y^{-1/2} - \log\det \mathbb{I}_d)
\end{aligned}$$

Let $\hat{a}_{y,i} := J_{y,i}^{-1/2} a_i / s_{y,i}$ for each $i \in [n]$. Using the property $\log\det B = \operatorname{trace} \log B$, where $\log B$ denotes the logarithm of the matrix and that $\log\det \mathbb{I}_d = 0$, we obtain

$$h^\top \log\det J_y = \lim_{\delta \to 0} \frac{1}{\delta} \left[\operatorname{trace} \log \left(\sum_{i=1}^{n} \frac{\zeta_{y+\delta h}}{(1 - \delta a_i^\top h / s_{y,i})} \hat{a}_{y,i} \hat{a}_{y,i}^\top \right) \right],$$

where we have substituted $s_{y+\delta h,i} = s_{y,i} - \delta a_i^\top h$. Keeping track of first order terms in δ, and noting that $\sum_{i=1}^{n} \zeta_{y,i} \hat{a}_{y,i} \hat{a}_{y,i}^\top = \mathbb{I}_d$, we find that

$$\begin{aligned}
&\operatorname{trace} \log \left(\sum_{i=1}^{n} \frac{\zeta_{y+\delta h,i}}{(1 - \delta a_i^\top h / s_{y,i})} \hat{a}_{y,i} \hat{a}_{y,i}^\top \right) \\
&= \operatorname{trace} \log \left[\sum_{i=1}^{n} \left(\zeta_{y,i} + \delta h^\top \nabla \zeta_{y,i} \right) \left(1 + \frac{2\delta a_i^\top h}{s_{y,i}} \right) \hat{a}_{y,i} \hat{a}_{y,i}^\top \right] + O\left(\delta^2\right) \\
&= \operatorname{trace} \left[\sum_{i=1}^{n} \delta \left(\frac{2a_i^\top h}{s_{y,i}} + h^\top \nabla \zeta_{y,i} \right) \hat{a}_{y,i} \hat{a}_{y,i}^\top \right] + O\left(\delta^2\right) \\
&= \sum_{i=1}^{n} \delta \left(\frac{2a_i^\top h}{s_{y,i}} + h^\top \nabla \zeta_{y,i} \right) \theta_{y,i} + O\left(\delta^2\right)
\end{aligned}$$

where in the last step we have used the fact that $\operatorname{trace}(\hat{a}_{y,i} \hat{a}_{y,i}^\top) = \hat{a}_{y,i}^\top \hat{a}_{y,i} = \theta_{y,i}$ for each $i \in [n]$. Substituting the expression for $\nabla \zeta_y$ from part (a), and rearranging the terms yields the claimed expression in the limit $\delta \to 0$.

Proof of part (d): Gradient of φ

Using the chain rule and the fact that $\nabla s_{y,i} = -a_i$, yields the result.

Proof of part (e)

We claim that

$$\begin{aligned}
&\frac{1}{2} h^\top \nabla^2 \Psi_y h \\
&= \frac{1}{2}\left[\sum_{i=1}^n \zeta_{y,i}\theta_{y,i}(3d_{y,i}^2 + 2d_{y,i}f_{y,i} + \ell_{y,i}) - \frac{1}{2}\sum_{i,j=1}^n \zeta_{y,i}\zeta_{y,j}\theta_{y,i,j}^2 \left(2d_{y,i} + f_{y,i}\right)\left(2d_{y,j} + f_{y,j}\right)\right].
\end{aligned}$$

The desired bound on $\left|h^\top \nabla^2 \Psi_y h\right|/2$ now follows from an application of AM-GM inequality with Lemma 32(d).

We now derive the claimed expression for the directional Hessian of the function Ψ. We have

$$\begin{aligned}
&\frac{1}{2} h^\top \left(\nabla^2 \log\det J_y\right) h \\
&= \lim_{\delta\to 0} \frac{1}{2\delta^2}(\log\det J_y^{-1/2} J_{y+\delta h} J_y^{-1/2} + \log\det J_y^{-1/2} J_{y-\delta h} J_y^{-1/2} - 2\log\det \mathbb{I}_d) \\
&= \frac{1}{2}\lim_{\delta\to 0}\frac{1}{\delta^2}\left[\operatorname{trace}\log\left(\sum_{i=1}^n \frac{\zeta_{y+\delta h}}{(1-\delta a_i^\top h/s_{y,i})}\hat{a}_{y,i}\hat{a}_{y,i}^\top\right) + \operatorname{trace}\log\left(\sum_{i=1}^n \frac{\zeta_{y-\delta h}}{(1+\delta a_i^\top h/s_{y,i})}\hat{a}_{y,i}\hat{a}_{y,i}^\top\right)\right].
\end{aligned}$$

Expanding the first term in the above expression, we find that

$$\begin{aligned}
&\operatorname{trace}\log\left(\sum_{i=1}^n \frac{\zeta_{y+\delta h,i}}{(1-\delta a_i^\top h/s_{y,i})}\hat{a}_{y,i}\hat{a}_{y,i}^\top\right) \\
&= \operatorname{trace}\log \underbrace{\left[\sum_{i=1}^n \left(\zeta_{y,i} + \delta h^\top \nabla\zeta_{y,i} + \frac{\delta^2}{2} h^\top \nabla^2 \zeta_{y,i} h\right)\left(1 + 2\delta\frac{a_i^\top h}{s_{y,i}} + 3\delta^2\frac{(a_i^\top h)^2}{s_{y,i}^2}\right)\hat{a}_{y,i}\hat{a}_{y,i}^\top\right]}_{=:\mathbb{I}_d + B} + O\left(\delta^3\right).
\end{aligned}$$

Substituting the shorthand notation from equations (B.58a), (B.58b) and (B.58c), we have

$$B = \sum_{i=1}^n \zeta_{y,i}\left[\delta(2d_{y,i} + f_{y,i}) + \delta^2(3d_{y,i}^2 + 2d_{y,i}f_{y,i} + \ell_{y,i})\right]\hat{a}_{y,i}\hat{a}_{y,i}^\top + O\left(\delta^3\right).$$

Now we make use of the following facts (1) $\operatorname{trace}\log(\mathbb{I}_d + B) = \operatorname{trace}\left[B - \frac{B^2}{2} + O\left(\|B\|^3\right)\right]$, (2) for each $i, j \in [n]$, we have $\operatorname{trace}(\hat{a}_{y,i}\hat{a}_j^\top) = \hat{a}_{y,i}^\top \hat{a}_j = \theta_{y,i,j}$, and (3) for each $i \in [n]$,

we have $\theta_{y,i,i} = \theta_{y,i}$. Thus we obtain

$$\begin{aligned}
&\operatorname{trace}\log\left(\sum_{i=1}^{n}\frac{\zeta_{y+\delta h,i}}{(1-\delta a_i^\top h/s_{y,i})}\hat{a}_{y,i}\hat{a}_{y,i}^\top\right)\\
&=\sum_{i=1}^{n}\zeta_{y,i}\theta_{y,i}\left[\delta(2d_{y,i}+f_{y,i})+\delta^2(3d_{y,i}^2+2d_{y,i}f_{y,i}+\ell_{y,i})\right]\\
&\quad-\frac{1}{2}\sum_{i,j=1}^{n}\zeta_{y,i}\zeta_{y,j}\theta_{y,i,j}^2\delta^2(2d_{y,i}+f_{y,i})(2d_{y,j}+f_{y,j})+O\left(\delta^3\right).
\end{aligned}$$

Similarly, we can obtain an expression for $\operatorname{trace}\log\left(\sum_{i=1}^{n}\frac{\zeta_{y-\delta h}}{(1+\delta a_i^\top h/s_{y,i})}\hat{a}_{y,i}\hat{a}_{y,i}^\top\right)$. Putting the pieces together, we obtain

$$\frac{1}{2}h^\top\left(\nabla^2\log\det J_y\right)h \tag{B.94}$$

$$=\sum_{i=1}^{n}\zeta_{y,i}\theta_{y,i}(3d_{y,i}^2+2d_{y,i}f_{y,i}+\ell_{y,i})-\frac{1}{2}\sum_{i,j=1}^{n}\zeta_{y,i}\zeta_{y,j}\theta_{y,i,j}^2(2d_{y,i}+f_{y,i})(2d_{y,j}+f_{y,j}). \tag{B.95}$$

Proof of part (f)

We claim that

$$\frac{1}{2}h^\top\nabla^2\varphi_{y,i}h=\varphi_{y,i}\left(2d_{y,i}f_{y,i}+3d_{y,i}^2+\ell_{y,i}\right). \tag{B.96}$$

The claim follows from a straightforward application of chain rule and substitution of the expressions for $\nabla\zeta_{y,i}$ and $\nabla^2\zeta_{y,i}$ in terms of the shorthand notation $d_{y,i}$, $f_{y,i}$ and $\ell_{y,i}$. Multiplying both sides of equation (B.96) with $d_{y,i}^2 s_{y,i}^2$ and summing over index i, we find that

$$\begin{aligned}
&\sum_{i=1}^{n}d_{y,i}^2s_{y,i}^2\frac{1}{2}h^\top\nabla\varphi_{y,i}^2h\\
&=\sum_{i=1}^{n}d_{y,i}^2s_{y,i}^2\varphi_{y,i}\left[\ell_{y,i}+2d_{y,i}f_{y,i}+3d_{y,i}^2\right]\\
&=\sum_{i=1}^{n}d_{y,i}^2\zeta_{y,i}\left[\ell_{y,i}+2d_{y,i}f_{y,i}+3d_{y,i}^2\right]\\
&\le\sum_{i=1}^{n}d_{y,i}^2\zeta_{y,i}\left[\ell_{y,i}+f_{y,i}^2+4d_{y,i}^2\right],
\end{aligned}$$

where in the last step we have used the AM-GM inequality. The claim follows.

B.8.2 Proof of Lemma 35

We claim that

$$0 \preceq G_y^{-1/2}\left(c_1\mathbb{I}_n + c_2\Lambda_y\left(G_y - \alpha\Lambda_y\right)^{-1}\right)G_y^{1/2} \preceq (c_1+c_2)\,\kappa\mathbb{I}_n. \tag{B.97}$$

The proof of the lemma is immediate from this claim, as for any PSD matrix $H \leq c\mathbb{I}_n$, we have $H^2 \leq c^2\mathbb{I}_n$.

We now prove claim (B.97). Note that

$$G_y^{-1/2}\Lambda_y\left(G_y - \alpha\Lambda_y\right)^{-1}G_y^{1/2} = \underbrace{G_y^{-1/2}\Lambda_y G_y^{-1/2}}_{:=B_y}(\mathbb{I}_n - \alpha_{\mathrm{J}}G_y^{-1/2}\Lambda_y G_y^{-1/2})^{-1}. \tag{B.98}$$

Note that the RHS is equal to the matrix $B_y(\mathbb{I}_n - \alpha_{\mathrm{J}}B_y)^{-1}$ which is symmetric. Observe the following ordering of the matrices in the PSD cone

$$\Sigma_y + \beta_{\mathrm{J}}\mathbb{I}_n = G_y \succeq \Sigma_y \succeq \Lambda_y = \Sigma_y - \Upsilon_y^{(2)} \succeq 0.$$

For the last step we have used the fact that $\Sigma_y - \Upsilon_y^{(2)}$ is a diagonally dominant matrix with non negative entries on the diagonal to conclude that it is a PSD matrix. Consequently, we have

$$B_y = G_y^{-1/2}\Lambda_y G_y^{-1/2} \preceq \mathbb{I}_n. \tag{B.99}$$

Further, recall that $\alpha_{\mathrm{J}} = (1 - 1/\kappa) \Leftrightarrow \kappa = (1-\alpha_{\mathrm{J}})^{-1}$. As s result, we obtain

$$0 \preceq (\mathbb{I}_n - \alpha_{\mathrm{J}}G_y^{-1/2}\Lambda_y G_y^{-1/2})^{-1} \preceq \kappa\mathbb{I}_n.$$

Multiplying both sides by $B_y^{1/2}$ and using the relation (B.99), we obtain

$$0 \preceq B_y^{1/2}(\mathbb{I}_n - \alpha_{\mathrm{J}}G_y^{-1/2}\Lambda_y G_y^{-1/2})^{-1}B_y^{1/2} \preceq \kappa\mathbb{I}_n. \tag{B.100}$$

Using the fact that B_y commutes with $(\mathbb{I}_n - B_y)^{-1}$, we obtain $B_y(\mathbb{I}_n - \alpha_{\mathrm{J}}B_y)^{-1} \preceq \kappa\mathbb{I}_n$. Using observation (B.98) now completes the proof.

B.8.3 Proof of Lemma 36

Without loss of generality, we can first prove the result for $i = 1$. Let $\nu := \mu_y^\top e_1$ denote the first row of the matrix μ_y. Observe that

$$e_1 = \left(G_y - \alpha\Lambda_y\right)G_y^{-1}\nu = \nu - \alpha\Sigma_y G_y^{-1}\nu + \alpha\Upsilon_y^{(2)}G_y^{-1}\nu \tag{B.101}$$

We now prove bounds (B.62a) and (B.62b) separately.

Proof of bound (B.62a)**:** Multiplying the equation (B.101) on the left by $\nu^\top G_y^{-1}$, we obtain

$$\begin{aligned} g_1^{-1}\nu_1 &= \nu^\top G_y^{-1}\nu - \alpha\nu^\top G_y^{-1}\Sigma_y G_y^{-1}\nu + \alpha\nu^\top G_y^{-1}\Upsilon_y^{(2)}G_y^{-1}\nu \\ &\geq \nu^\top G_y^{-1}\nu - \alpha\nu^\top G_y^{-1}\Sigma_y G_y^{-1}\nu \\ &\geq \left(g_1^{-1} - \alpha\sigma_{y,1}/g_1^2\right)\nu_1^2. \end{aligned} \tag{B.102}$$

Rearranging terms, we obtain

$$0 \leq \nu_1 \leq \frac{\zeta_{y,1}}{\zeta_{y,1} - \alpha\sigma_{y,1}} \overset{(i)}{\leq} \kappa, \tag{B.103}$$

where inequality (i) follows from the facts that $\zeta_{y,j} \geq \sigma_{y,j}$ and $(1-\alpha) = \kappa$.

Proof of bound (B.62b)**:** In our proof, we use the following improved lower bound for the term $\mu_{y,1,1} = \nu_1$.

$$\nu_1 \geq \frac{\zeta_{y,1}}{\zeta_{y,1} - \alpha\sigma_{y,1} + \alpha\sigma_{y,1}^2}, \tag{B.104}$$

Deferring the proof of this claim at the moment, we now complete the proof.

We begin by deriving a weighted ℓ_2-norm bound for the vector $\tilde{\nu} = (\nu_2, \ldots, \nu_n)^\top$. Equation (B.102) implies

$$\zeta_{y,1}^{-1}\nu_1\left(1 - \nu_1 + \alpha\frac{\sigma_{y,1}}{\zeta_{y,1}}\nu_1\right) \geq \sum_{j=2}^n \nu_j^2\left(\zeta_{y,j}^{-1} - \alpha\zeta_{y,j}^{-2}\sigma_{y,j}\right) \overset{(i)}{\geq} (1-\alpha)\sum_{j=2}^n \frac{\nu_j^2}{\zeta_{y,j}},$$

where step (i) follows from the fact that $\zeta_{y,i} \geq \sigma_{y,i}$. Now, we upper bound the expression on the left hand side of the above inequality using the upper (B.103) and lower (B.104) bounds on ν_1:

$$\begin{aligned} \zeta_{y,1}^{-1}\nu_1\left(1 - \nu_1 + \alpha\frac{\sigma_{y,1}}{\zeta_{y,1}}\nu_1\right) &\leq \zeta_{y,1}^{-1}\frac{\zeta_{y,1}}{\zeta_{y,1} - \alpha\sigma_{y,1}}\left(1 - \left(1 - \alpha\frac{\sigma_{y,1}}{\zeta_{y,1}}\right)\frac{\zeta_{y,1}}{\zeta_{y,1} - \alpha\sigma_{y,1} + \alpha\sigma_{y,1}^2}\right) \\ &= \frac{\alpha\sigma_{y,1}^2}{\left(\zeta_{y,1} - \alpha\sigma_{y,1}\right)\left(\zeta_{y,1} - \alpha\sigma_{y,1} + \alpha\sigma_{y,1}^2\right)} \\ &\leq \kappa^2, \end{aligned}$$

where in the last step we have used the facts that $\zeta_{y,1} \geq \sigma_{y,1}$ and $(1-\alpha)^{-1} = \kappa$. Putting the pieces together, we obtain $\sum_{j=2}^n \nu_j^2\zeta_{y,j}^{-1} \leq \kappa^3$, which is equivalent to our claim (B.62b) for $i = 1$.

It remains to prove our earlier claim (B.104). Writing equation (B.101) separately for the first coordinate and for the rest of the coordinates, we obtain

$$1 = \left(1 - \alpha\sigma_{y,1}\zeta_{y,1}^{-1} + \alpha\sigma_{y,1,1}^2\zeta_{y,j}^{-1}\right)\nu_1 + \alpha\sum_{j=2}^{n}\sigma_{y,1,j}^2\zeta_{y,j}^{-1}\nu_j, \quad \text{and} \tag{B.105a}$$

$$0 = \left(\mathbb{I}_{n-1} - \alpha\Sigma_y' G_y'^{-1}\right)\begin{pmatrix}\nu_2\\ \vdots\\ \nu_n\end{pmatrix} + \alpha\Upsilon_y'^{(2)}G_y'^{-1}\begin{pmatrix}\nu_2\\ \vdots\\ \nu_n\end{pmatrix} + \alpha\zeta_{y,1}^{-1}\nu_1\begin{pmatrix}\sigma_{y,1,2}^2\\ \vdots\\ \sigma_{y,1,n}^2\end{pmatrix}, \tag{B.105b}$$

where G_y' (respectively Σ_y', $\Upsilon_y'^{(2)}$) denotes the principal minor of G_y (respectively Σ_y, $\Upsilon_y^{(2)}$) obtained by excluding the first column and the first row. Multiplying both sides of the equation (B.105b) from the left by $\left(\nu_2, \cdots, \nu_n\right) G_y'^{-1}$, we obtain

$$0 = \sum_{j=2}^{n}\underbrace{\frac{1}{\zeta_{y,j}}\left(1 - \frac{\alpha\sigma_{y,j}}{\zeta_{y,j}}\right)}_{c_{y,j}}\nu_j^2 + \underbrace{\alpha\left(\nu_2, \cdots, \nu_n\right) G_y'^{-1}\Upsilon_y'^{(2)}G_y'^{-1}\begin{pmatrix}\nu_2\\ \vdots\\ \nu_n\end{pmatrix}}_{C_{y.2}} + \alpha\frac{\nu_1}{\zeta_{y,1}}\sum_{j=2}^{n}\frac{\sigma_{y,j}^2}{\zeta_{y,j}}\nu_j. \tag{B.106}$$

Observing that $\alpha \in [0,1]$ and $\zeta_{y,j} \geq \sigma_{y,j}$ for all $y \in \text{int}\left(\mathcal{K}\right)$ and $j \in [n]$, we obtain $c_{y,j} \geq 0$. Further, note that $G_y'^{-1}\Upsilon_y'^{(2)}G_y'^{-1}$ is a PSD matrix and hence we have that $C_{y,2} \geq 0$. Putting the pieces together, we have

$$\alpha\frac{\nu_1}{\zeta_{y,1}}\sum_{j=2}^{n}\frac{\sigma_{y,j}^2}{\zeta_{y,j}}\nu_j \leq 0.$$

Combining this inequality with equation (B.105a) yields the claim.

B.8.4 Proof of Corollary 8

Without loss of generality, we can prove the result for $i = 1$. Applying Cauchy-Schwarz inequality, we have

$$\|\nu\|_1 = \nu_1 + \sum_{j=2}^{n}|\nu_j| \leq \nu_1 + \sqrt{\sum_{j=2}^{n}\frac{\nu_j^2}{\zeta_{y,j}}\cdot\sum_{j=2}^{n}\zeta_{y,j}} \leq \kappa + \kappa^{3/2}\cdot\sqrt{1.5\ d} \leq 3\sqrt{d}\kappa^{3/2},$$

where to assert the last inequality we have used Lemma 36 and Lemma 28(c). The claim (B.63) follows. Further, noting that the infinity norm of a matrix is the ℓ_1-norm of its transpose, we obtain $\left\|\left(G_y - \alpha\Lambda_y\right)^{-1}G_y\right\|_\infty \leq 3\sqrt{d}\kappa^{3/2}$ as claimed.

B.9 Proof of Lemmas from Section B.5.2

In this section, we collect proofs of auxiliary lemmas from Section B.5.2.

B.9.1 Proof of Lemma 37

Using Lemma 33, and the relation (B.52) we have

$$\left(1 - \frac{s_{z,i}}{s_{x,i}}\right)^2 \leq 4\frac{r^2}{\kappa^4 d^{3/2}}\xi^\top \xi, \tag{B.107}$$

where $\xi \sim \mathcal{N}(0, \mathbb{I}_d)$. Define

$$\Delta_s := \max_{i\in[n],\ v\in\overline{xz}} \left|1 - \frac{s_{v,i}}{s_{x,i}}\right|. \tag{B.108}$$

Using the standard Gaussian tail bound, we observe that $\mathbb{P}_{\xi\sim\mathcal{N}(0,\mathbb{I}_n)}\left[\xi^\top\xi \geq d(1+\delta)\right] \leq 1 - \epsilon/4$ for $\delta = \sqrt{\frac{2}{d}}$. Plugging this bound in the inequality (B.107) and noting that for all $v \in \overline{xz}$ we have $\|v - x\|_{J_x} \leq \|z - x\|_{J_x}$, we obtain that

$$\mathbb{P}_{z\sim\mathcal{P}_x}\left[\Delta_s \leq \frac{2r^2(1+\sqrt{2/d}\log(4/\epsilon)}{\kappa^4\sqrt{d}}\right] \geq 1 - \epsilon/4.$$

Setting

$$r \leq 1/(25\sqrt{1+\sqrt{2}\log(4/\epsilon)}), \tag{B.109}$$

and noting that $\kappa^4\sqrt{d} \geq 1$ implies the claim (B.64a). Hence, we obtain that $\Delta_s < .005/\kappa^2$ and consequently $\max_{i\in[n],v\in\overline{xz}} s_{x,i}/s_{v,i} \in (0.99, 1.01)$ with probability at least $1 - \epsilon/4$.

We now claim that

$$\max_{i\in[n],v\in\overline{xz}} \frac{\zeta_{x,i}}{\zeta_{v,i}} \in \left[1 - 24\kappa^2\Delta_s, 1 + 24\kappa^2\Delta_s\right], \quad \text{if } \Delta_s \leq \frac{1}{32\kappa^2}.$$

The result follows immediately from this claim. To prove the claim, note that equation (B.68) implies that if $\Delta_s \leq \frac{1}{32\kappa^2}$, then

$$\frac{\zeta_{v,i}}{\zeta_{x,i}} \in (e^{-8\kappa^2\Delta_s}, e^{8\kappa^2\Delta_s}) \quad \text{for all } i \in [n] \text{ and } v \in \overline{xz},$$

which implies that

$$\max_{i\in[n],v\in\overline{xz}} \frac{\zeta_{x,i}}{\zeta_{v,i}} \in (e^{-8\kappa^2\Delta_s}, e^{8\kappa^2\Delta_s}).$$

Asserting the facts that $e^x \leq 1 + 3x$ and $e^{-x} \geq 1 - 3x$, for all $x \in [0, 1]$ yields the claim.

B.9.2 Proof of Lemma 38

The proof once again makes use of the classical tail bounds for polynomials in Gaussian random variables. We restate the classical result stated in equation (B.110) for convenience. For any $d \geq 1$, any polynomial $P : \mathbb{R}^d \to \mathbb{R}$ of degree k, and any $t \geq (2e)^{k/2}$, we have

$$\mathbb{P}\left[|P(\xi)| \geq t\left(\mathbb{E}P(\xi)^2\right)^{\frac{1}{2}}\right] \leq \exp\left(-\frac{k}{2e}t^{2/k}\right), \tag{B.110}$$

where $\xi \sim \mathcal{N}(0, \mathbb{I}_n)$ denotes a standard Gaussian vector in n dimensions.

Recall the notation from equation (B.60) and observe that

$$\|\hat{a}_{x,i}\|_2^2 = \theta_{x,i}, \quad \text{and} \quad \hat{a}_{x,i}^\top \hat{a}_{x,j} = \theta_{x,i,j}. \tag{B.111}$$

We also have

$$\sum_{i=1}^n \zeta_{x,i}\hat{a}_{x,i}\hat{a}_{x,i}^\top = J_x^{-1/2}\sum_{i=1}^n \zeta_{x,i}\frac{a_i a_i^\top}{s_{x,i}^2}J_x^{-1/2} = \mathbb{I}_d. \tag{B.112}$$

Further, using Lemma 35 we obtain

$$\sum_{i=1}^n \zeta_{x,i}\hat{b}_{x,i}\hat{b}_{x,i}^\top = J_x^{-1/2}A_x\Lambda_x\left(G_x - \alpha\Lambda_x\right)^{-1}G_x\left(G_x - \alpha\Lambda_x\right)^{-1}\Lambda_x A_x^\top J_x^{-1/2} = 4\kappa^2\mathbb{I}_d. \tag{B.113}$$

Throughout this section, we consider a fixed point $x \in \operatorname{int}(\mathcal{K})$. For brevity in our notation, we drop the dependence on x for terms like $\zeta_{x,i}, \theta_{x,i}, \hat{a}_{x,i}$ (etc.) and denote them simply by $\zeta_i, \theta_i, \hat{a}_i$ respectively.

We introduce some matrices and vectors that would come in handy for our proofs.

$$B = \begin{bmatrix}\sqrt{\zeta_1}\hat{a}_1^\top \\ \vdots \\ \sqrt{\zeta_n}\hat{a}_n^\top\end{bmatrix}, \quad B_b = \begin{bmatrix}\sqrt{\zeta_1}\hat{b}_1^\top \\ \vdots \\ \sqrt{\zeta_n}\hat{b}_n^\top\end{bmatrix}, \quad v = \begin{bmatrix}\sqrt{\zeta_1}\,\|\hat{a}_1\|_2^2 \\ \vdots \\ \sqrt{\zeta_n}\,\|\hat{a}_n\|_2^2\end{bmatrix}, \quad \text{and} \quad v^{ab} = \begin{bmatrix}\sqrt{\zeta_1}\hat{a}_1^\top\hat{b}_1 \\ \vdots \\ \sqrt{\zeta_n}\hat{a}_n^\top\hat{b}_n\end{bmatrix}. \tag{B.114}$$

We claim that

$$BB^\top \preceq \mathbb{I}_n, \quad \text{and} \quad B_b B_b^\top \preceq 4\kappa^2\mathbb{I}_n. \tag{B.115a}$$

To see these claims, note that equation (B.112) implies that $B^\top B = \mathbb{I}_d$ and consequently, $BB^\top$ is an orthogonal projection matrix and $BB^\top \preceq \mathbb{I}_n$. Next, note that from equation (B.113) we have that $B_b^\top B_b \preceq \kappa^2\mathbb{I}_d$, which implies that $B_b B_b^\top \preceq \kappa^2\mathbb{I}_n$. In asserting both these arguments, we have used the fact that for any matrix B, the matrices $BB^\top$ and $B^\top B$ are PSD and have same set of eigenvalues.

Next, we bound the ℓ_2 norm of the vectors v and v^{ab}:

$$\|v\|_2^2 = \sum_{i=1}^n \zeta_i \theta_i^2 \overset{\text{Lem. 32 (e)}}{\leq} 4d, \quad \text{and} \tag{B.115b}$$

$$\begin{aligned}\left\|v^{ab}\right\|_2^2 &= \sum_{i=1}^n \zeta_i \left(\hat{a}_i^\top \hat{b}_i\right)^2 \leq \sum_{i=1}^n \zeta_i \|\hat{a}_i\|_2^2 \left\|\hat{b}_i\right\|_2^2 \\ &\leq 4 \sum_{i=1}^n \zeta_i \left\|\hat{b}_i\right\|_2^2 = 4\,\mathrm{trace}(B_b^\top B_b) \overset{\text{eqn. (B.115a)}}{\leq} 16\kappa^2 d. \end{aligned} \tag{B.115c}$$

We now prove the five claims of the lemma separately.

Proof of bound (B.65a)

Using Isserlis' theorem [86] for fourth order Gaussian moments, we have

$$\mathbb{E}\left(\sum_{i=1}^n \zeta_i \left(\hat{a}_i^\top \xi\right)^2\right)^2 = \sum_{i,j=1}^n \zeta_i \zeta_j \left(\|\hat{a}_i\|_2^2 \|\hat{a}_j\|_2^2 + 2\left(\hat{a}_i^\top \hat{a}_j\right)^2\right) = \sum_{i,j=1}^n \zeta_i \zeta_j \left(\theta_i \theta_j + 2\theta_{i,j}^2\right) \leq 24 d^2,$$

where the last follows from Lemma 32. Applying the bound (B.110) with $k = 2$ and $t = e\log(\frac{16}{\epsilon})$. Note that the bound is valid since $t \geq (2e)$ for all $\epsilon \in (0, 1/30]$.

Proof of bound (B.65b)

Applying Isserlis' theorem for Gaussian moments, we obtain

$$\mathbb{E}\left(\sum_{i=1}^n \zeta_i \left(\hat{a}_i^\top \xi\right)^3\right)^2 = 9 \underbrace{\sum_{i,j=1}^n \zeta_i \zeta_j \|\hat{a}_i\|_2^2 \|\hat{a}_j\|_2^2 \left(\hat{a}_i^\top \hat{a}_j\right)}_{=:N_1} + 6 \underbrace{\sum_{i,j=1}^n \zeta_i \zeta_j \left(\hat{a}_i^\top \hat{a}_j\right)^3}_{=:N_2}.$$

We claim that $N_1 \leq 4d$ and $N_2 \leq 4d$. Assuming these claims as given at the moment, we now complete the proof. We have $\mathbb{E}\left(\sum_{i=1}^n \zeta_i \left(\hat{a}_i^\top \xi\right)^3\right)^2 \leq 60d$. Applying the bound (B.110) with $k = 3$ and $t = \left(\frac{2e}{3}\log\left(\frac{16}{\epsilon}\right)\right)^{3/2}$, and verifying that $t \geq (2e)^{3/2}$ for $\epsilon \in (0, 1/30]$ yields the claim.

We now turn to prove the bounds on N_1 and N_2. We have

$$\begin{aligned} N_1 &= \sum_{i,j=1}^n \zeta_i \|\hat{a}_i\|_2^2 \hat{a}_i^\top \zeta_j \|\hat{a}_j\|_2^2 \hat{a}_j \\ &= \left\|\sum_{i=1}^n \zeta_i \|\hat{a}_i\|_2^2 \hat{a}_i\right\|_2^2 = \left\|B^\top v\right\|_2^2 \overset{\text{eqn. (B.115a)}}{\leq} \|v\|_2^2 \overset{\text{eqn. (B.115b)}}{\leq} 4d. \end{aligned}$$

Next, applying Cauchy-Schwarz inequality and using equation (B.111), we obtain

$$N_2 = \sum_{i,j=1}^{n} \zeta_i \zeta_j \left(\hat{a}_i^\top \hat{a}_j\right)^3$$
$$\leq \sum_{i,j=1}^{n} \zeta_i \zeta_j \theta_{i,j}^2 \sqrt{\theta_i \theta_j} \overset{\text{(Lem. 28 (d))}}{\leq} 4 \sum_{i,j=1}^{n} \zeta_i \zeta_j \theta_{i,j}^2 \overset{\text{(Lem. 32 (d))}}{\leq} 4 \sum_{i=1}^{n} \zeta_i \theta_i = 4d.$$

Proof of bound (B.65c)

Using Isserlis' theorem for Gaussian moments, we have

$$\mathbb{E}\left(\sum_{i=1}^{n} \zeta_i \left(\hat{a}_i^\top \xi\right)^2 \left(\hat{b}_{x,i}^\top \xi\right)\right)^2 = \underbrace{\sum_{i,j=1}^{n} \zeta_i \zeta_j \left\|\hat{a}_i\right\|_2^2 \left\|\hat{a}_j\right\|_2^2 \left(\hat{b}_i^\top \hat{b}_j\right)}_{:=N_3} + 4 \underbrace{\sum_{i,j=1}^{n} \zeta_i \zeta_j \left(\hat{a}_i^\top \hat{a}_j\right) \left(\hat{a}_i^\top \hat{b}_i\right) \left(\hat{a}_j^\top \hat{b}_j\right)}_{:=N_4}$$
$$+ 4 \underbrace{\sum_{i,j=1}^{n} \zeta_i \zeta_j \left\|\hat{a}_i\right\|_2^2 \left(\hat{b}_i^\top \hat{a}_j\right) \left(\hat{a}_j^\top \hat{b}_j\right)}_{:=N_5} + 2 \underbrace{\sum_{i,j=1}^{n} \zeta_i \zeta_j \left(\hat{a}_i^\top \hat{a}_j\right)^2 \left(\hat{b}_i^\top \hat{b}_j\right)}_{:=N_6} + 4 \underbrace{\sum_{i,j=1}^{n} \zeta_i \zeta_j \left(\hat{a}_i^\top \hat{a}_j\right) \left(\hat{a}_i^\top \hat{b}_j\right) \left(\hat{b}_i^\top \hat{a}_j\right)}_{:=N_7}$$

We claim that all terms $N_k \leq 16\kappa^2 d, k \in \{3,4,5,6,7\}$. Putting the pieces together, we have

$$\mathbb{E}\left(\sum_{i=1}^{n} \zeta_i \left(\hat{a}_i^\top \xi\right)^2 \left(\hat{b}_{x,i}^\top \xi\right)\right)^2 \leq 240\kappa^2 d.$$

Applying the bound (B.110) with $k = 3$ and $t = \left(\frac{2e}{3} \log\left(\frac{16}{\epsilon}\right)\right)^{3/2}$ yields the claim. Note that for the given definition of t, we have $t \geq (2e)^{3/2}$ for $\epsilon \in (0, 1/30]$ so that the bound (B.110) is valid.

It is now left to prove the bounds on N_k for $k \in \{3,4,5,6,7\}$. We have

$$N_3 = \sum_{i,j=1}^{n} \zeta_i \|\hat{a}_i\|_2^2 \hat{b}_i^\top \zeta_j \left\|\hat{a}_j\right\|_2^2 \hat{b}_j = \left\|\sum_{i=1}^{n} \zeta_i \left\|\hat{a}_i\right\|_2^2 \hat{b}_i\right\|_2^2 = \left\|B_b^\top v\right\|_2^2 \overset{\text{eqn. (B.115a)}}{\leq} 4\kappa^2 \|v\|_2^2 \overset{\text{eqn. (B.115b)}}{=} \leq 16\kappa^2 d,$$
$$N_4 = \sum_{i,j=1}^{n} \zeta_i \zeta_j \left(\hat{a}_i^\top \hat{a}_j\right) \left(\hat{a}_i^\top \hat{b}_i\right) \left(\hat{a}_j^\top \hat{b}_j\right) = \left\|B^\top v^{ab}\right\|_2^2 \overset{\text{eqn. (B.115a)}}{\leq} \left\|v^{ab}\right\|_2^2 \overset{\text{eqn. (B.115c)}}{\leq} 16\kappa^2 d, \quad \text{and}$$
$$N_5 = \sum_{i,j=1}^{n} \zeta_i \zeta_j \left\|\hat{a}_i\right\|_2^2 \left(\hat{b}_i^\top \hat{a}_j\right) \left(\hat{a}_j^\top \hat{b}_j\right) = \left(B^\top v^{ab}\right)^\top \left(B_b^\top v\right) \overset{\text{C-S}}{\leq} \left\|B^\top v^{ab}\right\|_2 \left\|B_b^\top v\right\|_2 \leq 16\kappa^2 d.$$

For the term N_6, we have

$$\begin{aligned} N_6 = \sum_{i,j=1}^{n} \zeta_i \zeta_j \left(\hat{a}_i^\top \hat{a}_j\right)^2 \left(\hat{b}_i^\top \hat{b}_j\right) &\overset{\text{(C–S)}}{\leq} \frac{1}{2} \sum_{i,j=1}^{n} \zeta_i \zeta_j \left(\hat{a}_i^\top \hat{a}_j\right)^2 \left(\left\|\hat{b}_i\right\|_2^2 + \left\|\hat{b}_j\right\|_2^2\right) \\ &\overset{\text{(symm.in } i,j)}{=} \sum_{i,j=1}^{n} \zeta_i \zeta_j \left(\hat{a}_i^\top \hat{a}_j\right)^2 \left\|\hat{b}_i\right\|_2^2 \\ &\overset{\text{(eqn. (B.112))}}{\leq} \sum_{i=1}^{n} \zeta_i \left\|\hat{a}_i\right\|_2^2 \left\|\hat{b}_i\right\|_2^2 \\ &\overset{\text{(Lem. 28(}d\text{))}}{\leq} 4 \sum_{i=1}^{n} \zeta_i \left\|\hat{b}_i\right\|_2^2 \\ &\overset{\text{(eqn. (B.115c))}}{\leq} 16\kappa^2 d. \end{aligned}$$

The bound on the term N_7 can be obtained in a similar fashion.

Proof of bound (B.65d)

Observe that $\hat{a}_i^\top \xi \sim \mathcal{N}(0, \theta_i)$ and hence $\mathbb{E}\left(\hat{a}_i^\top \xi\right)^8 = 105\,\theta_i^4$. Thus, we have

$$\begin{aligned} &\mathbb{E}\left(\sum_{i=1}^{n} \zeta_i \left(\hat{a}_i^\top \xi\right)^4\right)^2 \\ &\overset{\text{C–S}}{\leq} \sum_{i,j=1}^{n} \zeta_i \zeta_j \left(\mathbb{E}\left(\hat{a}_i^\top \xi\right)^8\right)^{\frac{1}{2}} \left(\mathbb{E}\left(\hat{a}_j^\top \xi\right)^8\right)^{\frac{1}{2}} \\ &= 105 \sum_{i,j=1}^{n} \zeta_i \zeta_j \theta_i^2 \theta_j^2 \\ &= 105 \left(\sum_{i=1}^{n} \zeta_i \theta_i^2\right)^2. \end{aligned}$$

Now applying Lemma 32, we obtain that $\mathbb{E}\left(\sum_{i=1}^{n} \zeta_i \left(\hat{a}_i^\top \xi\right)^4\right)^2 \leq 1680 d^2$. Consequently, applying the bound (B.110) with $k = 4$ and $t = \left(\frac{e}{2} \log\left(\frac{16}{\epsilon}\right)\right)^2$ and noting that $t \geq (2e)^2$ for $\epsilon \in (0, 1/30]$, yields the claim.

Proof of bound (B.65e)

Using the fact that $\mathbb{E}\left(\hat{a}_i^\top \xi\right)^{12} = 945\,\theta_i^6$ and an argument similar to the previous part yields that $\mathbb{E}\left(\sum_{i=1}^{n} \zeta_i \left(\hat{a}_i^\top \xi\right)^6\right)^2 \leq 15120 d^2$.

Finally, applying the bound (B.110) with $k = 6$ and $t = \left(\frac{e}{3} \log\left(\frac{16}{\epsilon}\right)\right)^3$, and verifying that $t \geq (2e)^3$ for $\epsilon \in (0, 1/30]$, yields the claim.

B.10 Proof of Lovász's Lemma

We begin by formally defining the conductance (Φ) of a Markov chain on $(\mathcal{K}, \mathbb{B}(\mathcal{K}))$ with arbitrary transition operator $\mathcal{T}$ and stationary distribution Π^*. We assume that the operator $\mathcal{T}$ is lazy and thereby the stationary distribution Π^* is unique. Let $\mathcal{T}_x = \mathcal{T}(\delta_x)$ denote the transition distribution at point x, then the conductance Φ is defined as

$$\Phi := \inf_{\substack{S \in \mathbb{B}(\mathcal{K}) \\ \Pi^*(S) \in (0,1/2)}} \frac{\Phi(S)}{\Pi^*(S)} \quad \text{where} \quad \Phi(S) := \int_S \mathcal{T}_u(\mathcal{K} \cap S^c) d\Pi^*(u) \quad \text{for any } S \subseteq \mathcal{K}.$$

The conductance denotes the measure of the flow from a set to its complement relative to its own measure, when initialized in the stationary distribution. If the conductance is high, the following result shows that the Markov chain mixes fast.

Lemma 39. *[118, Theorem 1.4] For any M-warm start μ_0, the mixing time of the Markov chain with conductance Φ is bounded as*

$$\left\| \mathcal{T}^k(\mu_0) - \Pi^* \right\|_{TV} \leq \sqrt{M} \left(1 - \frac{\Phi^2}{2}\right)^k \leq \sqrt{M} \exp\left(-k \frac{\Phi^2}{2}\right).$$

Note that this result holds for a general distribution Π^* although we apply for uniform Π^*. The result can be derived from Cheeger's inequality for continuous-space discrete-time Markov chain and elementary results in Calculus. See, e.g., Theorem 1.4 and Corollary 1.5 by [118] for a proof. For ease in notation define $\mathcal{K} \backslash S := \mathcal{K} \cap S^c$. We now state a key isoperimetric inequality.

Lemma 40. *[115, Theorem 6] For any measurable sets $S_1, S_2 \subseteq \mathcal{K}$, we have*

$$\text{vol}(\mathcal{K} \backslash S_1 \backslash S_2) \cdot \text{vol}(\mathcal{K}) \geq d_{\mathcal{K}}(S_1, S_2) \cdot \text{vol}(S_1) \cdot \text{vol}(S_2),$$

where $d_{\mathcal{K}}(S_1, S_2) := \inf_{x \in S_1, y \in S_2} d_{\mathcal{K}}(x, y)$.

Since Π^* is the uniform measure on $\mathcal{K}$, this lemma implies that

$$\Pi^*(\mathcal{K} \backslash S_1 \backslash S_2) \geq d_{\mathcal{K}}(S_1, S_2) \cdot \Pi^*(S_1) \cdot \Pi^*(S_2). \tag{B.116}$$

In fact, such an inequality holds for an arbitrary log-concave distribution [121]. In words, the inequality says that for a bounded convex set any two subsets which are far apart, can not have a large volume. Taking these lemmas as given, we now complete the proof.

Proof of Lovász's Lemma: We first bound the conductance of the Markov chain using the assumptions of the lemma. From Lemma 39, we see that the Markov chain mixes fast if all the sets S have a high conductance $\Phi(S)$. We claim that

$$\Phi \geq \frac{\varrho \Delta}{64}, \tag{B.117}$$

from which the proof follows by applying Lemma 39. We now prove the claim (B.117) along the lines of Theorem 11 in the paper by [115]. In particular, we show that under the assumptions in the lemma, the sets with bad conductance are far apart and thereby have a small measure under Π^*, whence the ratio $\Phi(S)/\Pi^*(S)$ is not arbitrarily small. Consider a partition S_1, S_2 of the set $\mathcal{K}$ such that S_1 and S_2 are measurable. To prove claim (B.117), it suffices to show that

$$\frac{1}{\text{vol}(\mathcal{K})} \int_{S_1} \mathcal{T}_u(S_2) du \geq \frac{\varrho\Delta}{64} \cdot \min\left\{\Pi^*(S_1), \Pi^*(S_2)\right\}, \tag{B.118}$$

Define the sets

$$S_1' := \left\{ u \in S_1 \middle| \tilde{\mathcal{T}}_u(S_2) < \frac{\varrho}{2} \right\}, \quad S_2' := \left\{ v \in S_2 \middle| \tilde{\mathcal{T}}_v(S_1) < \frac{\varrho}{2} \right\}, \quad \text{and} \quad S_3' := \mathcal{K} \backslash S_1' \backslash S_2'. \tag{B.119}$$

Case 1: If we have $\text{vol}(S_1') \leq \text{vol}(S_1)/2$ and consequently $\text{vol}(\mathcal{K}\backslash S_1') \geq \text{vol}(S_1)/2$, then

$$\int_{S_1} \mathcal{T}_u(S_2) du \overset{(i)}{\geq} \frac{1}{2} \int_{S_1 \backslash S_1'} \tilde{\mathcal{T}}_u(S_2) du \overset{(ii)}{\geq} \frac{\varrho}{4} \text{vol}(S_1) \overset{(iii)}{\geq} \frac{\varrho\Delta}{4} \cdot \min\left\{\text{vol}(S_1), \text{vol}(S_2)\right\},$$

which implies the inequality (B.118) since Π^* is the uniform measure on $\mathcal{K}$. In the above sequence of inequalities, step (i) follows from the definition of the kernel $\mathcal{T}$, step (ii) follows from the definition of the set S_1' (B.119) and step (iii) from the fact that $\Delta < 1$. Dividing both sides by $\text{vol}(\mathcal{K})$ yields the inequality (B.118) and we are done.

Case 2: It remains to establish the inequality (B.118) for the case when $\text{vol}(S_i') \geq \text{vol}(S_i)/2$ for each $i \in \{1, 2\}$. Now for any $u \in S_1'$ and $v \in S_2'$ we have

$$\left\| \tilde{\mathcal{T}}_u - \tilde{\mathcal{T}}_v \right\|_{\text{TV}} \geq \tilde{\mathcal{T}}_u(S_1) - \tilde{\mathcal{T}}_v(S_1) = 1 - \tilde{\mathcal{T}}_u(S_2) - \tilde{\mathcal{T}}_v(S_1) > 1 - \varrho,$$

and hence by assumption we have $d_{\mathcal{K}}(S_1', S_2') \geq \Delta$. Applying Lemma 40 and the definition of S_3' (B.119) we find that

$$\text{vol}(S_3') \cdot \text{vol}(\mathcal{K}) \geq \Delta \cdot \text{vol}(S_1') \cdot \text{vol}(S_2') \geq \frac{\Delta}{4} \cdot \text{vol}(S_1) \cdot \text{vol}(S_2). \tag{B.120}$$

Using this inequality and the fact that $x(1-x) \geq \min\{x, (1-x)\}/2$ for any $x \in [0, 1]$, we obtain that

$$\Pi^*(S_3') \geq \frac{\Delta}{4} \cdot \Pi^*(S_1) \cdot \Pi^*(S_2) \geq \frac{\Delta}{8} \min\left\{\Pi^*(S_1), \Pi^*(S_2)\right\}. \tag{B.121}$$

We claim that

$$\int_{S_1} \mathcal{T}_u(S_2) du = \int_{S_2} \mathcal{T}_v(S_1) dv. \tag{B.122}$$

Assuming the claim as given, we now complete the proof. Using the equation (B.122), we have

$$\begin{aligned}\frac{1}{\mathrm{vol}(\mathcal{K})}\int_{S_1}\mathcal{T}_u(S_2)du &= \frac{1}{2\,\mathrm{vol}(\mathcal{K})}\left(\int_{S_1}\mathcal{T}_u(S_2)du+\int_{S_2}\mathcal{T}_v(S_1)dv\right)\\ &\overset{(i)}{\geq}\frac{1}{2\,\mathrm{vol}(\mathcal{K})}\left(\frac{1}{2}\int_{S_1\setminus S_1'}\tilde{\mathcal{T}}_u(S_2)du+\frac{1}{2}\int_{S_2\setminus S_2'}\tilde{\mathcal{T}}_v(S_2)dv\right)\\ &\overset{(ii)}{\geq}\frac{\varrho}{8}\frac{\mathrm{vol}(S_3')}{\mathrm{vol}(\mathcal{K})}\\ &\overset{(iii)}{\geq}\frac{\varrho\Delta}{64}\min\left\{\Pi^*(S_1),\Pi^*(S_2)\right\},\end{aligned}$$

where step (i) follows from the definition of the kernel $\mathcal{T}$, step (ii) follows from the definition of the set S_3' (B.119) and step (iii) follows from the inequality (B.121). Putting together the pieces yields the claim (B.117).

It remains to prove the claim (B.122). We make use of the following result

$$\Phi(S)=\Phi(\mathcal{K}\setminus S)\quad\text{for any measurable } S\subseteq\mathcal{K}. \tag{B.123}$$

Using equation (B.123) and noting that $S_1=\mathcal{K}\setminus S_2$, we have

$$\frac{1}{\mathrm{vol}(\mathcal{K})}\int_{S_1}\mathcal{T}_u(S_2)du=\int_{S_1}\mathcal{T}_u(S_2)\pi^*(u)du=\Phi(S_1)=\Phi(\mathcal{K}\setminus S_1)=\frac{1}{\mathrm{vol}(\mathcal{K})}\int_{S_2}\mathcal{T}_v(S_1)dv,$$

which yields equation (B.122).

Proof of result (B.123)**:** Note that $\int_{\mathcal{K}}\mathcal{T}_u(S)d\Pi^*(u)=\Pi^*(S)$. Thus, we have

$$\begin{aligned}\Phi(\mathcal{K}\setminus S)&=\int_{\mathcal{K}\setminus S}\mathcal{T}_u(S)d\Pi^*(u)\\ &=\int_{\mathcal{K}}\mathcal{T}_u(S)d\Pi^*(u)-\int_S\mathcal{T}_u(S)d\Pi^*(u)\\ &=\Pi^*(S)-\int_S\mathcal{T}_u(S)d\Pi^*(u).\end{aligned}$$

Using the fact that $1-\mathcal{T}_u(S)=\mathcal{T}_u(\mathcal{K}\setminus S)$, we obtain

$$\Pi^*(S)-\int_S\mathcal{T}_u(S)d\Pi^*(u)=\int_S d\Pi^*(u)-\int_S\mathcal{T}_u(S)d\Pi^*(u)=\int_S\mathcal{T}_u(\mathcal{K}\setminus S)d\Pi^*(u)=\Phi(S),$$

thereby yielding the claim (B.123).

Appendix C

Technical proofs for stability

C.1 Stability bounds for convex smooth functions

In this section, we prove stability bounds of optimization algorithms (GD, NAG and heavy ball methtod) for convex smooth functions.

Before we proceed to the main proof, we state several well known lemmas about convex optimization which can be found in [21, 24]. The L-smoothness of a function directly implies the following two lemmas. These two lemmas characterize how well the gradient approximation works for L-smooth functions in terms of both upper and lower bounds.

Lemma 41. *Let f be a L-smooth function on Ω. Then for all $u, v \in \Omega$, we have*

$$f(u) \leq f(v) + \nabla f(v)^\top (u - v) + \frac{L}{2} \|u - v\|_2^2$$

Lemma 42. *Let f be a convex and L-smooth function on Ω. Then for any $u, v \in \Omega$, we have*

$$f(u) \geq f(v) + \nabla f(v)^\top (u - v) + \frac{1}{2L} \|\nabla f(u) - \nabla f(v)\|_2^2$$

An immediate corollary could be obtained by applying from the Lemma 41 to (u, v) and then (v, u). This corollary directly implies the constracting property of the gradient decent method, which is the key component for providing its algorithmic uniform stability.

Corollary 9. *Let f be a L-smooth function on Ω. Then for any $u, v \in \Omega$, one has*

$$(\nabla f(u) - \nabla f(v))^\top (u - v) \geq \frac{1}{L} \|\nabla f(u) - \nabla f(v)\|_2^2$$

C.1.1 Gradient Descent

Recall that in order to prove the uniform stability, we need to bound the loss difference for any fixed sample z at each iteration $t \geq 1$

$$|l(\theta_t, z) - l(\theta'_t, z)|.$$

This quantity is related to the norm difference $\|\theta_t - \theta'_t\|_2$ under the M-Lipschitz condition. Using the update rule of full gradient method, we obtain an recursive relation on $\|\theta_t - \theta'_t\|_2$. For $\eta \leq \frac{1}{L}$ and $t \geq 1$, we have

$$\begin{aligned}
\|\theta_t - \theta'_t\|_2 &= \|\theta_{t-1} - \eta\nabla R_S(\theta_{t-1}) - \theta'_{t-1} + \eta\nabla R_{S'}(\theta'_{t-1})\|_2 \\
&\overset{(i)}{\leq} \left\|\theta_{t-1} - \theta'_{t-1} - \eta\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\theta_{t-1}) + \eta\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\theta'_{t-1})\right\|_2 + \frac{\eta}{n}\|\nabla f_k(\theta'_{t-1}) - \nabla f'_k(\theta'_{t-1})\|_2 \\
&\overset{(ii)}{\leq} \left\|\theta_{t-1} - \theta'_{t-1} - \eta\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\theta_{t-1}) + \eta\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\theta'_{t-1})\right\|_2 + \frac{2\eta M}{n} \\
&\overset{(iii)}{\leq} \|\theta_{t-1} - \theta'_{t-1}\|_2 + \frac{2\eta M}{n}
\end{aligned} \tag{C.1}$$

The inequality (i) uses triangular inequality. The inequality (ii) follows from the M-Lipschitz condition on the perturbed gradient terms. The last inequality (iii) is obtain via the contracting property of gradient descent proved in Lemma 42 and its Corollary 9.

Using the recursive relation, after summing Equation (C.1) from 1 to T, we prove that the fixed-step-size full gradient method at iteration T is $\frac{2\eta M^2 T}{n}$-uniform stable, for $\eta \leq \frac{1}{L}$. That is, for every $z \in \mathcal{Z}$,

$$|l(\theta_T; z) - l(\theta'_T; z)| \leq \frac{2\eta M^2 T}{n}.$$

We remark that the stability of fixed-step-size full gradient method is linear as a function of iteration T. More generally, for gradient descent with varying step-sizes, using the same arguments, we can prove that the stability is upper bounded by the cumulative sum of all previous step-sizes at T.

Next, we show that this stability upper bound can be achieved by a linear function. We design the loss function $l(\theta; z)$ such that it is either $M\theta$ or $-M\theta$ depending on z. We define the two empirical loss functions on S and S',

$$\begin{aligned}
R_S(\theta) &= \frac{1}{n}\sum_{j=1}^{n} M\theta = M\theta, \\
R_{S'}(\theta) &= -\frac{1}{n}M\theta + \frac{1}{n}\sum_{j=1, j\neq k}^{n} M\theta = \frac{n-2}{n}M\theta.
\end{aligned}$$

The two empirical loss functions differ exactly by $\frac{2}{n}M\theta$. We have for iteration T,

$$\theta_T = T\eta M + \theta_0,$$
$$\theta'_T = \frac{n-2}{n}T\eta M + \theta_0$$

Then for this linear loss, for any $z \in \mathcal{Z}$,

$$|l(\theta_T; z) - l(\theta'_T; z)| = \frac{2\eta M^2 T}{n}.$$

The stability upper bound is thus tight.

C.1.2 Nesterov's Accelerated Gradient Descent

Recall that the Nesterov's accelerated gradient method has the following updates for $t \geq 1$:

$$\theta_{t+1} = (1-\gamma_{t-1})\,\theta_t + \gamma_{t-1}\theta_{t-1} - \eta\nabla R_S((1-\gamma_{t-1})\,\theta_t + \gamma_{t-1}\theta_{t-1}), \tag{C.2}$$

where $\eta \leq \frac{1}{L}$ is the step-size. γ_t is defined by the following recursion

$$\lambda_0 = 0, \lambda_t = \frac{1+\sqrt{1+4\lambda_{t-1}^2}}{2}, \text{ and } \gamma_t = \frac{1-\lambda_t}{\lambda_{t+1}},$$

satisfying $-1 < \gamma_t \leq 0$. For the updates on the perturbed samples S', we have

$$\theta'_{t+1} = (1-\gamma_{t-1})\,\theta'_t + \gamma_{t-1}\theta'_{t-1} - \eta\nabla R_{S'}((1-\gamma_{t-1})\,\theta'_t + \gamma_{t-1}\theta'_{t-1}). \tag{C.3}$$

Denote $\Delta\theta_t = \theta_t - \theta'_t$. Taking the difference of Equation (C.2) and (C.3), we have

$$\Delta\theta_{t+1} = (1-\gamma_{t-1})\,\Delta\theta_t + \gamma_{t-1}\Delta\theta_{t-1} - \eta\nabla^2 R_S(\theta_{\text{mid},t})\,((1-\gamma_{t-1})\,\Delta\theta_t + \gamma_{t-1}\Delta\theta_{t-1}) + e_t. \tag{C.4}$$

where the error term satisfies

$$e_t = \eta\nabla R_{S'}((1-\gamma_{t-1})\,\theta'_t + \gamma_{t-1}\theta'_{t-1}) - \eta\nabla R_S((1-\gamma_{t-1})\,\theta'_t + \gamma_{t-1}\theta'_{t-1}),$$

and $\theta_{\text{mid},t}$ is on the path from $(1-\gamma_{t-1})\,\theta_t + \gamma_{t-1}\theta_{t-1}$ to $(1-\gamma_{t-1})\,\theta'_t + \gamma_{t-1}\theta'_{t-1}$. Note that we have used the mean value theorem to group two gradient terms.

Because $\nabla R_{S'}$ and ∇R_S only differ in one term, using the M-Lipschitz gradient property, we obtain an upper bound on the error term

$$\|e_t\|_2 \leq \frac{2\eta M}{n}.$$

In the case of quadratic objective, we can denote

$$A = \eta \nabla^2 R_S(\theta_{\text{mid},t}).$$

Using the convex and L-smooth property, we have

$$0 \preceq A \preceq \mathbb{I}_d.$$

Then we can rewrite Equation C.4 as follows,

$$\Delta\theta_{t+1} = (\mathbb{I}_d - A)\left[(1-\gamma_{t-1})\,\Delta\theta_t + \gamma_{t-1}\Delta\theta_{t-1}\right] + e_t.$$

Writing this in matrix form, we have

$$\begin{pmatrix} \Delta\theta_{t+1} \\ \Delta\theta_t \end{pmatrix} = \begin{pmatrix} (1-\gamma_{t-1})\,(\mathbb{I}_d - A) & \gamma_{t-1}\,(\mathbb{I}_d - A) \\ \mathbb{I}_d & 0 \end{pmatrix} \begin{pmatrix} \Delta\theta_t \\ \Delta\theta_{t-1} \end{pmatrix} + \begin{pmatrix} e_t \\ 0 \end{pmatrix}. \tag{C.5}$$

Denote $G_t = \begin{pmatrix} (1-\gamma_{t-1})\,(\mathbb{I}_d - A) & \gamma_{t-1}\,(\mathbb{I}_d - A) \\ \mathbb{I}_d & 0 \end{pmatrix}$. Then we have an explicit expression of $\Delta\theta_{t+1}$ by applying the update equation (C.5) recursively, for $t \geq 1$,

$$\begin{pmatrix} \Delta\theta_{t+1} \\ \Delta\theta_t \end{pmatrix} = \prod_{i=1}^{t} G_i \begin{pmatrix} \Delta\theta_1 \\ \Delta\theta_0 \end{pmatrix} + \sum_{i=0}^{t-1} \prod_{s=t-i+1}^{t} G_s \begin{pmatrix} e_{t-i} \\ 0 \end{pmatrix}. \tag{C.6}$$

We have used $\prod_{i=1}^{t} G_i$ to denote the matrix product $G_t G_{t-1} \ldots G_1$. The goal is to bound the norm of $\Delta\theta_{t+1}$. We will need the following lemma on the spectral norm of $\prod_{i=1}^{t} G_i$ to conclude.

Lemma 43. *Suppose* $M_t = \begin{pmatrix} (1-\gamma_t)B & \gamma_t B \\ 1 & 0 \end{pmatrix}$, *where* $B \in \mathbb{R}^{d\times d}$ *is a symmetric positive semi-definite matrix* $0 \preceq B \preceq \mathbb{I}_d$ *and* $-1 < \gamma_t < 1$. *Then for all* $t \geq 1$,

$$\left\| \prod_{i=1}^{t} M_i \right\|_2 \leq 2(t+1).$$

Assuming Lemma 43 as given at the moment, we now complete the proof. According to Equation (C.6), applying Lemma 43 to G_t, we can bound the norm of $\Delta\theta_{t+1}$,

$$\begin{aligned} \|\Delta\theta_{t+1}\|_2 &\leq 2(t+1)\frac{2\eta M}{n} + \sum_{i=0}^{t-1} 2(i+1)\frac{2\eta M}{n} \\ &= \frac{2\eta M}{n}\left(t^2 + 3t + 1\right) \\ &\leq \frac{4\eta M}{n}\left(t+1\right)^2. \end{aligned}$$

We have used the fact that $\|\Delta\theta_0\|_2 = 0$, $\|\Delta\theta_1\|_2 \leq \frac{2\eta M}{n}$ and $\|e_t\|_2 \leq \frac{2\eta M}{n}$ in the first inequality. Together with the M-Lipschitz condition, we obtain that the Nesterov accelerated gradient method at iteration T is

$$\frac{4\eta M^2 T^2}{n}$$

uniform stable.

Now we turn back to prove Lemma 43.

Proof of Lemma 43 Since B is symmetric positive-semidefinite, we can diagonalize B. There exists a common orthogonal matrix Q and diagonal matrices D such that

$$B = Q^{-1} D Q.$$

We have $0 \preceq D \preceq \mathbb{I}_d$. As a consequence, M_i could also be decomposed as follows,

$$M_i = \begin{pmatrix} Q^{-1} & 0 \\ 0 & Q^{-1} \end{pmatrix} \begin{pmatrix} (1-\gamma_{i-1})\, D & \gamma_{i-1} D \\ \mathbb{I}_d & 0 \end{pmatrix} \begin{pmatrix} Q & 0 \\ 0 & Q \end{pmatrix}.$$

Then we obtain for its product

$$\prod_{i=1}^{t} M_i = \begin{pmatrix} Q^{-1} & 0 \\ 0 & Q^{-1} \end{pmatrix} \left[\prod_{i=1}^{t} \begin{pmatrix} (1-\gamma_{i-1})\, D & \gamma_{i-1} D \\ \mathbb{I}_d & 0 \end{pmatrix} \right] \begin{pmatrix} Q & 0 \\ 0 & Q \end{pmatrix}.$$

We observe that $\left[\prod_{i=1}^{t} \begin{pmatrix} (1-\gamma_{i-1})\, D & \gamma_{i-1} D \\ \mathbb{I}_d & 0 \end{pmatrix} \right]$ is a block diagonal matrix. To bound the spectral norm of $\left[\prod_{i=1}^{t} \begin{pmatrix} (1-\gamma_{i-1})\, D & \gamma_{i-1} D \\ \mathbb{I}_d & 0 \end{pmatrix} \right]$, it is sufficient to bound the 2×2 matrix of the following form

$$\prod_{i=1}^{t} H_i,$$

where

$$H_i = \begin{pmatrix} (1-\gamma_{i-1})h & \gamma_{i-1} h \\ 1 & 0 \end{pmatrix},$$

with $0 \leq h \leq 1$. To bound its spectral norm, we claim the following lemma.

Lemma 44. *Suppose* $H_i = \begin{pmatrix} (1-\gamma_{i-1})h & \gamma_{i-1} h \\ 1 & 0 \end{pmatrix}$, *where* $0 \leq h \leq 1$ *and* $-1 < \gamma_{i-1} < 1$. *Then*

$$\left\| \prod_{i=1}^{t} H_i \right\|_2 \leq 2(t+1).$$

Assuming Lemma 44 as given at the moment, the Lemma 43 can be completed.

$$\left\|\prod_{i=1}^{t} G_i\right\|_2 \leq \left\|\left[\prod_{i=1}^{t}\begin{pmatrix}(1-\gamma_{i-1})\,D & \gamma_{i-1}D\\ \mathbb{I}_d & 0\end{pmatrix}\right]\right\|_2 \leq 2(t+1).$$

Now we turn back to prove Lemma 44.

Proof of Lemma 44 Note that $\prod_{i=1}^{t} H_i$ is a 2×2 matrix. Let $\begin{pmatrix}a_0\\ b_0\end{pmatrix}$ be a vector with norm 1. We define

$$\begin{pmatrix}a_t\\ b_t\end{pmatrix} = \prod_{i=1}^{t} H_i \begin{pmatrix}a_0\\ b_0\end{pmatrix}.$$

To bound the spectral norm of $\prod_{i=1}^{t} H_i$, it is sufficient to bound the norm of $\begin{pmatrix}a_t\\ b_t\end{pmatrix}$. We going to show by recursion that

$$\max\left(|a_t|, |b_t|\right) \leq 2(t+1).$$

For $t=0, t=1$, the statement is easy to verify.
Suppose that the statement is true until t. We have the following recursion,

$$\begin{aligned} a_{t+1} &= h\left((1-\gamma_t)a_t + \gamma_t b_t\right)\\ b_{t+1} &= a_t. \end{aligned}$$

We remark that a_{t+1} as a function of $(\gamma_0, \ldots, \gamma_t)$ is a multivariate polynomial with degree one. Hence its maximum or minimum value is attained at the extreme values of the variables. Formally,

$$|a_{t+1}| \leq \max_{(\gamma_i)_{0\leq i\leq t}\in\{-1,1\}^{t+1}} |a_{t+1}(\gamma_0, \ldots, \gamma_t)|$$

This is a combinatorial optimization problem. But we observe that there are only four relevant cases.

- If $\gamma_t = 1$, then we have

$$\begin{aligned} a_{t+1} &= hb_t\\ b_{t+1} &= a_t. \end{aligned}$$

  Applying the assumption of the recursion, we obtain the desired bound for a_{t+1} and b_{t+1}.

- If $\gamma_1 = 1$, then we have

$$a_1 = hb_0$$
$$b_1 = a_0.$$

$\begin{pmatrix} a_1 \\ b_1 \end{pmatrix}$ is a vector with norm less than 1. Consider the problem with $\begin{pmatrix} a_1 \\ b_1 \end{pmatrix}$ as initialization, we obtain the desired bound for a_{t+1} and b_{t+1}.

- If there exists $i \in \{2, \ldots, t-1\}$ such that $\gamma_i = 1$, then

$$H_i = \begin{pmatrix} 0 & h \\ 1 & 0 \end{pmatrix},$$

and

$$H_{i+1}H_iH_{i-1} = h \begin{pmatrix} (1 - \gamma_{i+1} + \gamma_{i+1}(1 - \gamma_{i-1}))\, h & \gamma_{i+1}\gamma_{i-1}h \\ 1 & 0 \end{pmatrix},$$

Since $-1 \leq \gamma_{i+1}\gamma_{i-1} \leq 1$, this problem is again reduced to the problem where only $t-2$ matrices are multiplied together: from H_t to H_{i+2}, then $H_{i+1}H_iH_{i-1}$, then from H_{i-2} to H_1. We apply the assumption of the recursion and obtain the desired bound for a_{t+1}.

- Otherwise, all $\gamma_0, ..., \gamma_t$ should take value -1. Then

$$H_i = \begin{pmatrix} 2h & -h \\ 1 & 0 \end{pmatrix}.$$

Let $\begin{pmatrix} H_t^{11} & H_t^{12} \\ H_t^{21} & H_t^{22} \end{pmatrix} = \prod_{i=1}^{t} H_i$, then we have the following recursion for its entries

$$\begin{aligned} H_{i+1}^{11} &= 2hH_i^{11} - hH_i^{21}, \\ H_{i+1}^{21} &= H_i^{11}, \\ H_{i+1}^{12} &= 2hH_i^{12} - hH_i^{22}, \\ H_{i+1}^{22} &= H_i^{12}. \end{aligned}$$

We note that H_i^{11} satisfies the following second-order recursion

$$H_{i+1}^{11} = 2hH_i^{11} - hH_{i-1}^{11},$$

with $H_0^{11} = 1$ and $H_0^{11} = 2h$. We observe that H_i^{11} is exactly the Chebyshev polynomial [185, 131] of second kind with parameter $U_i(h)$. It is known that for Chebyshev polynomial of second kind,

$$U_i(\cos(\theta)) = \frac{\sin((i+1)\theta)}{\sin(\theta)},$$

and if $z = e^{i\theta}$,

$$\begin{aligned} |U_i(\cos(\theta))| &= \left| \frac{z^{i+1} - z^{-i-1}}{z - z^{-1}} \right| \\ &= \left| z^{-2i} \right| \left| \sum_{j=0}^{i} z^{2j} \right| \\ &\leq i + 1. \end{aligned}$$

Thus

$$\left| H_{t+1}^{11} \right| \leq t + 2.$$

Similarly, we show that all entries are less than $t + 2$. As a consequence,

$$\max(|a_{t+1}|, |b_{t+1}|) \leq 2(t + 2).$$

This discussion of four relevant cases concludes the recursion part, and thus the proof of Lemma 44.

C.1.3 Heavy Ball Method with Fixed Momentum

The proof of the fixed momentum heavy ball method proceeds similarly to that of the Nesterov accelerated gradient descent.

Fixed momentum heavy ball method has the following updates.

$$\theta_{t+1} = \theta_t - \eta \nabla R_{S'}(\theta_t) + \gamma (\theta_t - \theta_{t-1}), \tag{C.7}$$

with fixed momentum $\gamma \in [0, 1)$, and fixed step-size $\eta \in \left(0, \frac{(1-\gamma)}{L}\right)$. For the updates on the perturbed samples S', we have

$$\theta'_{t+1} = \theta'_t - \eta \nabla R_{S'}(\theta'_t) + \gamma \left(\theta'_t - \theta'_{t-1}\right). \tag{C.8}$$

Denote $\Delta\theta_t = \theta_t - \theta'_t$. Taking the difference of Equation (C.7) and (C.8), we have

$$\Delta\theta_{t+1} = (1 + \gamma)\Delta\theta_t - \gamma\Delta\theta_{t-1} - \eta \nabla^2 R_S(\theta_{\text{mid},t})(\Delta\theta_t) + e_t, \tag{C.9}$$

where the error term satisfies

$$e_t = \eta \nabla R_{S'}(\theta'_t) - \eta \nabla R_S(\theta'_t),$$

and $\theta_{\text{mid},t}$ is on the path from θ_t to θ'_t. Here we have used the mean value theorem to group the two gradient terms and to make appear the Hessian terms. Using the M-Lipschitz property, we obtain an upper bound on the error term,

$$\|e_t\|_2 \leq \frac{2\eta M}{n}.$$

In the case of quadratic objective, we can denote

$$A = \eta \nabla^2 R_S(\theta_{\text{mid},t}).$$

Using the convex and L-smooth property, we have

$$0 \preceq A \preceq \eta L \mathbb{I}_d.$$

We can rewrite Equation (C.9) in matrix form,

$$\begin{pmatrix} \Delta\theta_{t+1} \\ \Delta\theta_t \end{pmatrix} = \begin{pmatrix} (1+\gamma)\,\mathbb{I} - A & -\gamma\mathbb{I} \\ \mathbb{I} & 0 \end{pmatrix} \begin{pmatrix} \Delta\theta_t \\ \Delta\theta_{t-1} \end{pmatrix} + \begin{pmatrix} e_t \\ 0 \end{pmatrix} \tag{C.10}$$

Denote $G = \begin{pmatrix} (1+\gamma)\mathbb{I}_d - A & -\gamma\mathbb{I}_d \\ \mathbb{I}_d & 0 \end{pmatrix}$. Then we could obtain an explicit expression for the difference term as follows,

$$\begin{pmatrix} \Delta\theta_{t+1} \\ \Delta\theta_t \end{pmatrix} = \prod_{i=1}^{t} G_i \begin{pmatrix} \Delta\theta_1 \\ \Delta\theta_0 \end{pmatrix} + \sum_{i=0}^{t-1} \prod_{s=t-i+1}^{t} G_s \begin{pmatrix} e_{t-i} \\ 0 \end{pmatrix}. \tag{C.11}$$

As in the proof of NAG in Appendix C.1.2, we are going to bound the spectral norm of $\prod_{i=1}^{t} G_i$ to conclude. Using diagonalization of the matrices A, it is sufficient to consider products of the 2×2 matrices $H = \begin{pmatrix} 1+\gamma-a & -\gamma \\ 1 & 0 \end{pmatrix}$, with $0 \le a \le \eta L$. The following lemma characterizes the spectral norm of $\prod_{i=1}^{t} H$.

Lemma 45. *Suppose* $H = \begin{pmatrix} 1+\gamma-a & -\gamma \\ 1 & 0 \end{pmatrix}$, *where* $0 < \gamma < 1$ *and* $0 \le a \le 1-\gamma$. *Then*

$$\left\| \prod_{i=1}^{t} H \right\|_2 \le \frac{2}{1-\sqrt{\gamma}}.$$

Assuming Lemma 45 as given at the moment, we have

$$\left\| \prod_{i=1}^{t} G_i \right\|_2 \le \frac{2}{1-\sqrt{\gamma}}.$$

We can complete the proof of Theorem 11.

$$\begin{aligned} \|\Delta\theta_{t+1}\|_2 &\le \frac{2}{1-\sqrt{\gamma}} \frac{2\eta M}{n} + \sum_{i=0}^{t-1} \frac{2}{1-\sqrt{\gamma}} \frac{2\eta M}{n} \\ &= \frac{4\eta M}{(1-\sqrt{\gamma})n} (t+1). \end{aligned}$$

We have used the fact that $\|\Delta\theta_0\|_2 = 0$, $\|\Delta\theta_1\|_2 \leq \frac{2\eta M}{n}$ and $\|e_t\|_2 \leq \frac{2\eta M}{n}$ in the first inequality. Together with the M-Lipschitz condition, we obtain that the heavy ball method with fixed momentum at iteration T is

$$\frac{4\eta M^2 T}{(1-\sqrt{\gamma})n}$$

uniform stable.

Now we turn back to prove Lemma 45.

Proof of Lemma 45 Let $\prod_{i=1}^{t} H = \begin{pmatrix} a_t & b_t \\ c_t & d_t \end{pmatrix}$. We are going to show by recursion that

$$\max(|a_t|, |b_t|, |c_t|, |d_t|) \leq \frac{1}{1-\sqrt{\gamma}}.$$

For $t = 0, 1$, the statement is easy to verify.
Suppose that the statement is true until t. We have by recursion formular

$$\begin{aligned} a_{t+1} &= ((1+\gamma-a)a_t - \gamma c_t) \\ c_{t+1} &= a_t \\ b_{t+1} &= ((1+\gamma-a)b_t - \gamma d_t) \\ d_{t+1} &= b_t \end{aligned}$$

with initialization $a_1 = 1+\gamma-a, c_1 = 1, b_1 = -\gamma, d_1 = 0$. We remark that a_i satisfies the following second-order recursion, for $i \geq 1$,

$$a_{i+1} = (1+\gamma-a)a_i - \gamma a_{i-1},$$

where $a_0 = 1, a_1 = 1+\gamma-a$. We can also add $a_{-1} = 0$.

The characteristic equation is

$$x^2 - (1+\gamma-a)\,x + \gamma = 0.$$

The two roots are

$$x_{1,2} = \frac{1+\gamma-a \pm \sqrt{(1+\gamma-a)^2 - 4\gamma}}{2}.$$

We note that

$$|x_{1,2}| \leq 1.$$

. We distinguish two cases based on the two roots.

- The two roots are distinct. By distinct roots theorem for second order homogeneous system, we have

$$a_t = l_1 x_1^{t+1} + l_2 x_2^{t+1},$$

where l_1 and l_2 are constants to be determined by the initial condition. Solving the initial condtion, we have

$$l_1 = \frac{1}{\sqrt{(1+\gamma-a)^2 - 4\gamma}}$$
$$l_2 = -\frac{1}{\sqrt{(1+\gamma-a)^2 - 4\gamma}}.$$

Hence, we can bound a_t as follows,

$$\begin{aligned}
|a_t| &\leq \frac{1}{\left|\sqrt{(1+\gamma-a)^2 - 4\gamma}\right|} |x_1 - x_2| \left|\sum_{i=0}^{t} x_1^{t-i} x_2^i\right| \\
&\leq \sum_{i=0}^{t} |x_2|^i \\
&\leq \sum_{i=0}^{t} \sqrt{\gamma}^i \\
&\leq \frac{1}{1-\sqrt{\gamma}}.
\end{aligned}$$

We have used that $|x_2| \leq \sqrt{\gamma}$. When the two roots have imaginary part, it is clear that $|x_2| = \sqrt{\gamma}$. On the other hand, when the two roots are real, since $|x_1 x_2| = \gamma$, $|x_2| \leq |x_1|$, we also have $|x_2| \leq \sqrt{\gamma}$.

- The two roots are equal. $1 + \gamma - a = 2\sqrt{\gamma}$.

$$x_{1,2} = \sqrt{\gamma} < 1$$

By single root theorem for second order homogeneous system, We have

$$a_t = (1+t)\sqrt{\gamma}^t \leq \sum_{i=0}^{t} \sqrt{\gamma}^t \leq \frac{1}{1-\sqrt{\gamma}}.$$

Overall, we have proved a bound for a_t,

$$|a_t| \leq \frac{1}{1-\sqrt{\gamma}}.$$

We can bound b_t, c_t and d_t similarly because they have similar recursion formular.

$$\max(|a_t|, |b_t|, |c_t|, |d_t|) \leq \frac{1}{1-\sqrt{\gamma}}.$$

Using the relationship between spectral norm and Frobenius norm, we have

$$\left\| \prod_{i=1}^{t} H \right\|_2 \leq \frac{2}{1-\sqrt{\gamma}}.$$

C.2 Stability bounds for strongly convex smooth functions

C.2.1 Gradient Descent

Recall that in order to prove the uniform stability, we need bound the loss difference for any fixed sample z at each iteration $t \geq 1$

$$|l(\theta_t, z) - l(\theta'_t, z)|.$$

This quantity is related to the norm difference $\|\theta_t - \theta'_t\|_2$ under the M-Lipschitz condition. Under m-strongly-convex case, we bound $\|\theta_t - \theta'_t\|_2$ slightly different than that in the convex smooth case.

Using the update rule of full gradient method, we obtain an recursive relation on $\|\theta_t - \theta'_t\|_2$. For $\eta \leq \frac{2}{m+L}$ and $t \geq 1$, we have

$$\begin{aligned}
\|\theta_t - \theta'_t\|_2 &= \left\|\theta_{t-1} - \eta \nabla R_S(\theta_{t-1}) - \theta'_{t-1} + \eta \nabla R_{S'}(\theta'_{t-1})\right\|_2 \\
&\overset{(i)}{\leq} \left\|\theta_{t-1} - \theta'_{t-1} - \eta \nabla R_S(\theta_{t-1}) + \eta \nabla R_S(\theta'_{t-1})\right\|_2 + \frac{\eta}{n}\left\|\nabla f_k(\theta'_{t-1}) - \nabla f'_k(\theta'_{t-1})\right\|_2 \\
&\overset{(ii)}{\leq} \left\|\theta_{t-1} - \theta'_{t-1} - \eta \nabla R_S(\theta_{t-1}) + \eta \nabla R_S(\theta'_{t-1})\right\|_2 + \frac{2\eta M}{n} \\
&\overset{(iii)}{\leq} \left(1 - \frac{2mL\eta}{m+L}\right)^{1/2} \left\|\theta_{t-1} - \theta'_{t-1}\right\|_2 + \frac{2\eta M}{n} \\
&\overset{(iv)}{\leq} \left(1 - \frac{mL\eta}{m+L}\right) \left\|\theta_{t-1} - \theta'_{t-1}\right\|_2 + \frac{2\eta M}{n} \qquad \text{(C.12)} \\
& \qquad \text{(C.13)}
\end{aligned}$$

The inequality (i) uses triangular inequality. The inequality (ii) follows from the M-Lipschitz condition on the perturbed gradient terms. The inequality (iii) is obtain via the following claim, for f m-strongly convex and L-smooth, we have

$$(\nabla f(x) - \nabla f(y))^\top (x - y) \geq \frac{mL}{m+L}\|x - y\|_2^2 + \frac{1}{m+L}\|\nabla f(x) - \nabla f(y)\|_2^2. \qquad \text{(C.14)}$$

This claim can be easily obtain by plugging $f(x) - \frac{m}{2}\|x\|_2^2$, which is a convex function into Corollary 9. The inequality (iv) uses the fact $(1-x)^{1/2} \leq 1 - x^{1/2}$, for $0 \leq x \leq 1$.

Using the recursive relation, after summing Equation (C.12) from 1 to T, we have

$$\begin{aligned}\|\theta_t - \theta_t'\|_2 &\leq \frac{2\eta M}{n}\left(\sum_{i=0}^{T-1}\left(1 - \frac{mL\eta}{m+L}\right)^i\right) \\ &= \frac{4M}{mn}\left(1 - \left(1 - \frac{\eta L}{1+\kappa}\right)^T\right).\end{aligned}$$

Applying the M-Lipschitz condition, we have for every $z \in \mathcal{Z}$,

$$|l(\theta_T; z) - l(\theta_T'; z)| \leq \frac{4M^2}{mn}\left(1 - \left(1 - \frac{\eta L}{1+\kappa}\right)^T\right).$$

C.2.2 Nesterov's Accelerated Gradient Descent

According to the discussion of Equation C.5, in the case of quadratic loss, the Nesterov accelerated gradient descent difference term is as follows

$$\begin{pmatrix}\Delta\theta_{t+1} \\ \Delta\theta_t\end{pmatrix} = \begin{pmatrix}(1+\gamma)(\mathbb{I}_d - A) & -\gamma(\mathbb{I}_d - A) \\ \mathbb{I}_d & 0\end{pmatrix}\begin{pmatrix}\Delta\theta_t \\ \Delta\theta_{t-1}\end{pmatrix} + \begin{pmatrix}e_t \\ 0\end{pmatrix},$$

where

$$\gamma = \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1},$$

$m\eta\mathbb{I}_d \leq A \leq L\eta\mathbb{I}_d$ and $\|e_t\|_2 \leq \frac{2\eta M}{n}$.

Denote $G = \begin{pmatrix}(1+\gamma)(\mathbb{I}_d - A) & -\gamma(\mathbb{I}_d - A) \\ \mathbb{I}_d & 0\end{pmatrix}$. Then we could obtain an explicit expression for the difference term as follows,

$$\begin{pmatrix}\Delta\theta_{t+1} \\ \Delta\theta_t\end{pmatrix} = \prod_{i=1}^{t} G_i \begin{pmatrix}\Delta\theta_1 \\ \Delta\theta_0\end{pmatrix} + \sum_{i=0}^{t-1}\prod_{s=t-i+1}^{t} G_s \begin{pmatrix}e_{t-i} \\ 0\end{pmatrix}. \tag{C.15}$$

As in the proof of NAG in Appendix C.1.2, we are going to bound the spectral norm of $\prod_{i=1}^{t} G_i$ to conclude. Following the proof idea used in Appendix C.1.2 and Appendix C.1.3, using diagonalization of the matrices A, it is sufficient to consider products of the 2×2 matrices $H = \begin{pmatrix}(1+\gamma)h & -\gamma h \\ 1 & 0\end{pmatrix}$, with $1 - L\eta \leq h \leq 1 - m\eta$. The following lemma characterizes the spectral norm of $\prod_{i=1}^{t} H$.

Lemma 46. *Suppose* $H = \begin{pmatrix} (1+\gamma)h & -\gamma h \\ 1 & 0 \end{pmatrix}$, *where* $\gamma = \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}$ *and* $1 - L\eta \leq h \leq 1 - m\eta$. *Then*

$$\left\| \prod_{i=1}^{t} H \right\|_2 \leq 2(1+t) \left(\gamma(1-m\eta)\right)^{t/2}.$$

Assuming Lemma 46 as given at the moment, we have

$$\left\| \prod_{i=1}^{t} G_i \right\|_2 \leq 2(1+t) \left(\gamma(1-m\eta)\right)^{t/2}.$$

We can complete the proof of Theorem 13.

$$\begin{aligned} \|\Delta\theta_{t+1}\|_2 &\leq \frac{2\eta M}{n} \left(2(1+t) \left(\gamma(1-m\eta)\right)^{t/2} + \sum_{i=0}^{t-1} 2(1+i) \left(\gamma(1-m\eta)\right)^{i/2} \right) \\ &= \frac{4\eta M}{n} \left(\sum_{i=0}^{t} (1+i) \left(\gamma(1-m\eta)\right)^{i/2} \right) \end{aligned}$$

We have used the fact that $\|\Delta\theta_0\|_2 = 0$, $\|\Delta\theta_1\|_2 \leq \frac{2\eta M}{n}$ and $\|e_t\|_2 \leq \frac{2\eta M}{n}$ in the first inequality. Let $p = \left(\gamma(1-m\eta)\right)^{1/2}$ and

$$S = \sum_{i=0}^{t} (1+i) p^i.$$

Then

$$(1-p)S = \sum_{i=0}^{t} p^i - (t+1)p^{t+1} \leq \frac{1-p^{t+1}}{1-p}.$$

We also have upper and lower bounds on p,

$$p^2 = \gamma(1-m\eta) = \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1} \cdot \frac{\kappa - \eta L}{\kappa} \leq \left(\frac{\sqrt{\kappa} - \sqrt{\eta L}}{\sqrt{\kappa}} \right)^2,$$

and

$$p^2 \geq \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}} \right)^2.$$

Thus

$$\begin{aligned} \|\Delta\theta_{t+1}\|_2 &\leq \frac{4\eta M}{n} \left(\sum_{i=0}^{t} (1+i) \left(\gamma(1-m\eta)\right)^{i/2} \right) \\ &\leq \frac{4\eta M}{(1-p)^2 n} \left(1 - p^{t+1}\right) \\ &\leq \frac{4M}{mn} \left(1 - \left(1 - \frac{1}{\sqrt{\kappa}} \right)^{t+1} \right). \end{aligned}$$

Together with the M-Lipschitz condition, we obtain that the heavy ball method with fixed momentum at iteration T is

$$\frac{4M^2}{mn}\left(1-\left(1-\frac{1}{\sqrt{\kappa}}\right)^T\right)$$

uniform stable.

Now we turn back to prove Lemma 46.

Proof of Lemma 46 Let $\prod_{i=1}^{t} H = \begin{pmatrix} a_t & b_t \\ c_t & d_t \end{pmatrix}$. We are going to show by recursion that

$$\max(|a_t|,|b_t|,|c_t|,|d_t|) \leq (1+t)\left(\gamma(1-m\eta)\right)^{t/2}.$$

For $t = 0, 1$, the statement is easy to verify.
Suppose that the statement is true until t. We have by recursion formular

$$\begin{aligned} a_{t+1} &= ((1+\gamma)ha_t - \gamma h c_t) \\ c_{t+1} &= a_t \\ b_{t+1} &= ((1+\gamma)hb_t - \gamma h d_t) \\ d_{t+1} &= b_t \end{aligned}$$

with initialization $a_1 = (1+\gamma)h, b_1 = -\gamma h, c_1 = 1$ and $d_1 = 0$. We remark that a_i, satisfies the following second-order recursion, for $i \geq 1$,

$$a_{i+1} = (1+\gamma)ha_i - \gamma h a_{i-1},$$

where $a_0 = 1, a_1 = (1+\gamma)h$. We can also add $a_{-1} = 0$.

The characteristic equation is

$$x^2 - (1+\gamma)hx + \gamma h = 0.$$

The two roots are

$$x_{1,2} = \frac{(1+\gamma)h \pm \sqrt{(1+\gamma)^2h^2 - 4\gamma h}}{2}.$$

We verify that

$$\Delta = (1+\gamma)^2h^2 - 4\gamma h = 4h\left(\frac{\kappa h - (\kappa-1)}{(\sqrt{\kappa}+1)^2}\right) \leq 0,$$

because $h \leq 1 - m\eta \leq \frac{\kappa-1}{\kappa}$. Hence either we have equal real roots, or we have complex roots with imaginary parts.

We distinguish these two cases.

- The two roots are equal. $(1+\gamma)h = 2\sqrt{\gamma h}$. Then

$$x_{1,2} = \sqrt{\gamma h} < 1.$$

By single root theorem for second order homogeneous system, we have

$$a_t = (1+t)\left(\gamma h\right)^{t/2} \leq (1+t)\left(\gamma(1-m\eta)\right)^{t/2}.$$

- The two roots are distinct.

$$|x_{1,2}| = \sqrt{\gamma h} < 1.$$

By distinct roots theorem for second order homogeneous system, we have

$$a_t = l_1 x_1^{t+1} + l_2 x_2^{t+1},$$

where l_1 and l_2 are constants to be determined by the initial condition. Solving the initial condtion, we have

$$\begin{aligned} l_1 &= \frac{1}{\sqrt{(1+\gamma)^2 h^2 - 4\gamma h}} \\ l_2 &= -\frac{1}{\sqrt{(1+\gamma)^2 h^2 - 4\gamma h}}. \end{aligned}$$

Hence, we can bound a_t as follows,

$$\begin{aligned} |a_t| &\leq \frac{1}{\left|\sqrt{(1+\gamma)^2 h^2 - 4\gamma h}\right|} |x_1 - x_2| \left|\sum_{i=0}^{t} x_1^{t-i} x_2^i\right| \\ &\leq \sum_{i=0}^{t} \left(\gamma h\right)^{t/2} \\ &\leq (1+t)\left(\gamma(1-m\eta)\right)^{t/2}. \end{aligned}$$

We can bound b_t, c_t and d_t similarly because they have similar recursion formular.

$$\max(|a_t|, |b_t|, |c_t|, |d_t|) \leq (1+t)\left(\gamma(1-m\eta)\right)^{t/2}..$$

Using the relationship between spectral norm and Frobenius norm, we have

$$\left\|\prod_{i=1}^{t} H\right\|_2 \leq 2(1+t)\left(\gamma(1-m\eta)\right)^{t/2}.$$

Appendix D

Support information for DeepTune

D.1 Data collection

Extracellular recordings were made from well isolated neurons in parafoveal areas V4 (71 neurons) of three awake, behaving male rhesus macaques (Macaca mulatta). This dataset has been previously used to study the sparseness of neural codes in the area V4 [197]. Surgical procedures are thus identical to those in [197]. We restate the procedures here for completeness. Surgical procedures were conducted under appropriate anesthesia using standard sterile techniques [194]. Areas V4 were located by exterior cranial landmarks and/or direct visualization of the lunate sulcus, and location was confirmed by comparing receptive field properties and response latencies to those reported previously [66, 175].

During recording, the animals performed a fixation task for a liquid reward. Eye position was monitored with an infrared eye tracker (500 Hz; Eyelink II; SR Research) and trials during which eye position deviated $> 0.5°$ from the fixation spot were excluded from our analysis. The standard deviation of the fixational eye movements was typically $0.1°$. Activity was recorded using tungsten electrodes (FHC), and amplified and neural signals were isolated using a spike sorter (Plexon).

Experiments were controlled and stimuli generated using custom behavioral/stimulus display software (PyPE) running on a Linux-based PC. Stimuli were displayed on a 21 inch Trinitron monitor (Sony) capable of displaying luminances up to $500 Cd/m^2$. The luminance nonlinearity (gamma) of the monitor was calibrated and corrected in software to provide a linear luminance response.

In the main experiment, each neuron was probed with a rapidly changing sequence of natural images. The images were circular patches of grayscale digital photographs from a commercial digital library (Corel). Patches were chosen by an automated algorithm that selected them at random but favored patches with high contrast [to reduce the frequency of blank stimuli (e.g., patches of sky)]. All patches were adjusted with a gamma nonlinearity of 2.2, to give an appropriate luminance profile on our linearized display. The outer edges of the patches (10% of the radius) were blended smoothly

into the neutral gray background, whose luminance was chosen to match the mean luminance of the image sequence.

Random images were then concatenated into long sequences so that each 16.7 ms frame contained a random image patch from the library. All images were centered on the classical receptive field (CRF) and patch size was adjusted to be two to four times the CRF diameter. The entire sequence was broken into 3–5 s segments, and one segment was presented on each fixation trial. To avoid transient trial onset effects, the first 196 ms of data acquired on each trial were discarded before analysis.

The training dataset of a neuron consists of $4,000 - 12,000$ natural images. Spike count was measured at 60 Hz, resulting in two measurements per image. For the holdout dataset, 300 images were shown for each neuron, distinct from the images shown for the training dataset. The sequence of test images was repeated; on average, each image in the test set was shown 9.3 times. The resulting spike counts were averaged to provide a lower-variance estimate of the expected spike count; repeats also allowed for estimation of the amount variance in the neuron explainable by the stimulus image (signal-to-noise).

Figure D.1. Sample of Images from training and holdout datasets. A. 50 images sampled from training dataset of 4000 images of one neuron. B. 25 images sampled from holdout dataset of 300 images of one neuron.

D.1.1 Classical receptive field (CRF) estimation

After isolating each neuron, the boundaries of the classical receptive field (CRF) were estimated using bars and gratings. The CRF was localized precisely by reverse correlation of responses to a dynamic sparse noise stimulus: black and white squares or bars positioned randomly on a gray background and randomly repositioned at $5 - 10$ Hz [93, 47, 194]. The bars were scaled so that six to eight squares spanned the manually estimated receptive field ($0.1 - 1.5°$/square). The CRF was defined as the circle around the region where sparse noise stimulation elicited spiking responses. Our manual and automatic estimation procedures were generally in good agreement. CRF diameters ranged

from 0.5 to 10.4° (median, 2.2°), and eccentricities ranged from 0.1 to 49° (median, 3.1°).

D.1.2 Repeats in the holdout test set for explainable variance estimation

The use of repeats and the explainable variance estimation follows [176]. As it is explained in [176], even a perfect model cannot make perfect predictions, because the neuronal response has a non-deterministic component. Even if the model was completely identical to the neuron in every aspect, it would nevertheless be unable to explain 100% of the variance in the neuronal responses because the responses collected over two separate sets of stimulus presentations cannot be expected to be identical and the first set does not perfectly predict the second. A good measure of model performance for sensory neural systems should take these considerations into account and judge model performance relative to achievable, rather than total, prediction accuracy.

Specifically, for M repeated trials, we denote R_m as the recording firing rate from the m-th repeat. Also denote $\hat{y}$ as the predicted firing rate and y as the average recorded firing rate.

$$y(t) = \frac{1}{M}\sum_{m=1}^{M} R_m(t).$$

We want a measure of performance that characterize the similarity between predicted firing rate $\hat{y}$ and the recorded one y better than the simple correlation. For this, the correlation coefficient $\mathrm{CC}_{\mathrm{abs}}$, the noised corrected correlation coefficient $\mathrm{CC}_{\mathrm{norm}}$, the signal power (SP) and the total power (TP) are defined as follows,

$$\begin{aligned}
\mathrm{CC}_{\mathrm{abs}} &= \frac{\mathrm{Cov}(y,\hat{y})}{\sqrt{\mathrm{Var}(y)\,\mathrm{Var}(\hat{y})}} \\
\mathrm{CC}_{\mathrm{norm}} &= \frac{\mathrm{Cov}(y,\hat{y})}{\sqrt{\mathrm{Var}(\hat{y})\cdot \mathrm{SP}}} \\
\mathrm{SP} &= \frac{\mathrm{Var}\left(\sum_{m=1}^{M} R_m\right) - \sum_{m=1}^{M}\mathrm{Var}(R_m)}{M(M-1)} \\
\mathrm{TP} &= (M-1)\cdot\sum_{m=1}^{M}\mathrm{Var}(R_m).
\end{aligned}$$

While TP has an unexplainable part, SP is explainable in principle by a model. $\mathrm{CC}_{\mathrm{norm}}$ thus aims to quantify model performance relative to the best achievable performance. We use $\mathrm{CC}^2_{\mathrm{norm}}$ as the explainbale variance estimate.

D.2 Methods

In this section, we discuss the methods we used to model single neurons in V4 and the relevant metrics to measure the performance of our models.

D.2.1 Single neuron modeling and metrics

As described in the main text, we use transfer learning framework to analyze sinlge V4 Neuron input-response data: We first extract convolutional neural networks (CNN) features and then use as predictors in a linear regression method to predict spike-rates as the response. The CNNs are pretrained on large scale image classification dataset ImageNet [172]. The linear model learned by regularized linear regression is trained on our data.

As a measure of the prediction performance of our model, the correlation between the expected spike count predicted by the model and the actual average spike count on the holdout set is computed.

Explainable variance captured by the model is another relevant metric for predction performance in the neuroscience litterature [168, 176]. This metric attempts to control for differences in noise levels between experimental setups, individual neurons, and brain regions. As we explain in the last section, we estimate explainable variance using the repeat presentations of images in the test set. We use CC^2_{norm} as an estimate of the explainable variance.

D.2.2 Convolutional neural networks (CNN)

Deep convolutional neural networks are a successful tool to analyze big data problems and are therefore are being actively studied for a vast variety of applications especially in machine learning [105, 101, 177].

Convolutional networks are basically neural networks with several layers and a specialized connectivity structure. The purpose is to extract features of the scene in multiple layers. It has been shown that higher layers compute more global features than lower layers, so that the hierarchical structure provides a better overall quality of features [208]. The proposed architecture for several layers of network varies in different applications but it usually consists of three general types: convolutional layers, pooling layers and fully-connected layers.

Convolutional layers select a window of previous layer's output and convolve it with a set of filters. Dependencies are local in this structure. The coefficients of these filters are tunable weights of our network and their final value will be specified in the training procedure. As an example, considering images as the input of our network, each filter is a rectangular grid which will be convolved with specific patch of previous layer. A non-linear function will be used to specify the output of the neuron as in traditional neural networks. Equation D.1 specifies this relationship between output of different layers for two-dimensional configuration.

$$y_{i,j}^{l} = f\left(\sum_{m=0}^{M}\sum_{n=0}^{M} w_{mn} y_{i+m,j+n}^{l-1}\right) \tag{D.1}$$

where i and j's indicates possible spatial location at layer l, $y_{i,j}^{l}$ is the output of each neuron in layer l, w_{mn} is the filter weights at location (m, n) of layer $l-1$ and f is the non-linear function.

Pooling layers could be utilized after each convolutional layer. It simply performs a spatial pooling over patches of previous layer. These patches could be overlapping or non-overlapping. The output of pooling layer for each patch is a single value which in most of the cases is maximum value of the patch. Pooling could be useful to reduce the feature dimension as well as increase the invariance of the features for small transformation. It also helps to increase the size of receptive field for each feature value.

After several convolutional and pooling layers aimed at grasping the low-level and high-level features, a few *fully-connected layers* are used as the final stages of the network. These layers are essential for specific application of the network such as classification or prediction.

Figure D.2 shows the neural network architecture of the AlexNet model [101]. It consists of five convolutional layers, three pooling layers inbetween and two fully connected layers. Our analysis is carried out on all the seven layers shown in the figures. Layers L2, L3 and L4 are of the main focus. In particular, the output feature at L2 is of size $256 \times 13 \times 13$, where 256 indicates the number of types of filters applied at Layer L2, 13×13 indicates that the features are extracted on a spatially-equally-spaced 13×13 overlapping grid of the original image. Similarly the output features at layers L3 and L4 are of size $384 \times 13 \times 13$ and $384 \times 13 \times 13$.

We also used GoogLeNet [183] and VGG [179] in our analysis. The architectures and the mechanism of these models are beyond the scope. We refer the readers to the original paper for a detailed understanding. The CNN feature extraction pipeline is done using the Caffe [90] package and the model files provided within.

D.2.3 Regression methods

As described in the main text, our predictive model for a single neuron response takes the following form

$$\begin{aligned} F : \mathbb{R}^{s\times s\times k} &\to \mathbb{R} \\ (\mathbf{z}_t, ..., \mathbf{z}_{t-k+1}) &\mapsto \sum_{j=0}^{k-1} \boldsymbol{\beta}_{j+1}^{\top} h(\mathbf{z}_{t-j}), \end{aligned}$$

where $(\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_k) \in \mathbb{R}^{d\times k}$ are the regression parameters to be determined and h is the fixed CNN feature transform.

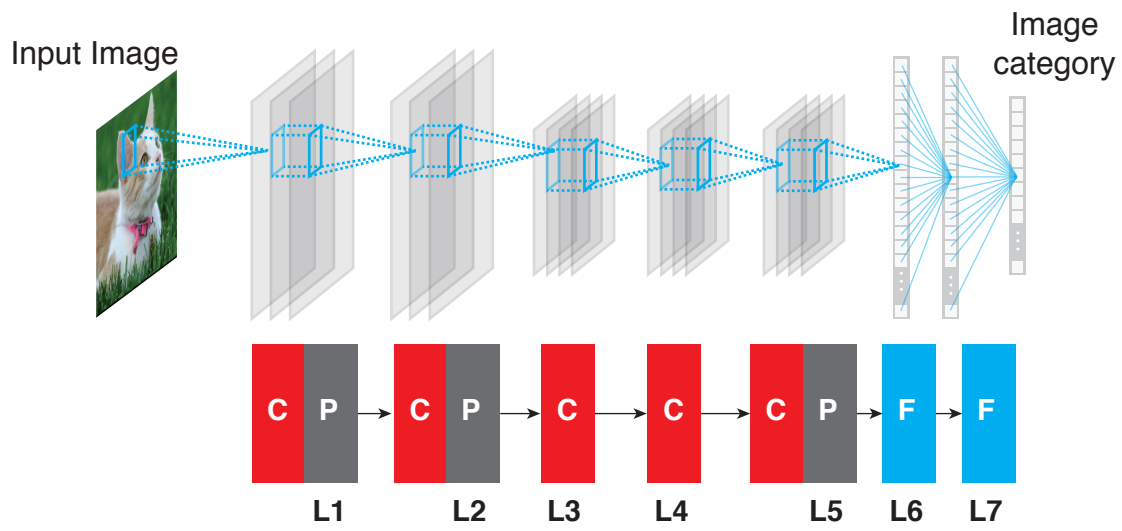


Figure D.2. Architecture of the AlexNet model [101]. Red box indicates convolutional layer, gray box indicates pooling layer and blue box indicates fully connected layer.

To perform the regression analysis, we solve the following regularized linear regression problem

$$\left(\hat{\boldsymbol{\beta}}_1, \ldots, \hat{\boldsymbol{\beta}}_k\right) = \underset{\boldsymbol{\beta}_1,\ldots,\boldsymbol{\beta}_k}{\arg\min} \frac{1}{2} \sum_{t=k}^{T} \left(y_t - \sum_{j=0}^{k-1} \boldsymbol{\beta}_{j+1}^{\top} h(\mathbf{z}_{t-j}) \right)^2 + \lambda_1 \sum_{j=1}^{k} \left\|\boldsymbol{\beta}_j\right\|_1 + \lambda_2 \sum_{j=1}^{k} \left\|\boldsymbol{\beta}_j\right\|_2^2 .$$

Taking the AlexNet Layer 2 model as an example, the Layer 2 feature is of dimension $d = 256 \times 13 \times 13$. Taking into account the time lags, the weight matrix $\left(\hat{\boldsymbol{\beta}}_1, \ldots, \hat{\boldsymbol{\beta}}_k\right)$ is of dimension $256 \times 13 \times 13 \times 9$. This feature dimension is much larger than the sample size $T = 8000$. Regularization methods are needed to both improve prediction accuracy and provide better interpretation.

Either ridge regression (ℓ_2 regularization) or LASSO [186] (ℓ_1 regularization) will be suitable for this high dimensional regression problem. While ridge regression is commonly used in the neuroscience litterature, LASSO could provide better guarantees for feature selection [209] in theory. We find that both regression methods produce consistent prediction performance and DeepTune images in our analysis. A detailed comparison is discussed in Section D.3.

D.2.4 DeepTune image generation

For a given model (e.g. AlexNet-Layer2+Ridge), the DeepTune image is defined as one image that maximize the model response under the constraints that it is smooth and naturalistic. Specifically, given the model $f : \mathbb{R}^{s \times s} \mapsto \mathbb{R}$, we seek an input image

$\mathbf{z} \in \mathbb{R}^{s \times s}$ that minimizes the following objective function:

$$-f(\mathbf{z}) + \lambda_p \mathcal{R}_p(\mathbf{z}) + \lambda_{\text{TV}} \mathcal{R}_{\text{TV}}(\mathbf{z}). \tag{D.2}$$

The regularization terms $\mathcal{R}_p$ and $\mathcal{R}_{\text{TV}}$ are motivated by image denoising techniques [171] and by natural image statistics [178, 125]. They are defined as follows,

$$\mathcal{R}_p(\mathbf{z}) = \sum_{i=1}^{s} \sum_{j=1}^{s} |\mathbf{z}\,(i,j)|^p\,,$$
$$\mathcal{R}_{\text{TV}}\,(\mathbf{z}) = \sum_{i=1}^{s} \sum_{j=1}^{s} \left[(\mathbf{z}\,(i,j+1) - \mathbf{z}\,(i,j))^2 + [\mathbf{z}\,(i+1,j) - \mathbf{z}\,(i,j)]^2 \right]^{\frac{1}{2}},$$

where $\mathbf{z}\,(i,j)$ is the pixel value of the image stimulus $\mathbf{z}$ at location (i,j).

In addition to that, balancing loss and regulariser(s) by choosing λ_p and λ_{TV} requires some attention. The optimal tuning for one neuron is achieved by cross-validation on a training set split (90% training + 10% validation). Then parameter tuning (via binary search) is stopped when we observe visually that the DeepTune is smooth enough and also contains enough details. This parameter setting is fixed for all other 70 neurons once we have tuned them for one neuron. This is to avoid over-tuning these parameters, which could result in over-interpretation.

D.2.5 Consensus DeepTune image generation

The goal of the consensus DeepTune image is to aggregate all 18 DeepTune images by keeping only the stable parts of them. The consensus DeepTune image is obtained via a similar optimization scheme as in the original DeepTune optimization for a single model in Equation (D.2). But insteading of using one gradient, an aggregation of gradient information from all 18 models is used. Let $f_i : \mathbb{R}^{s \times s} \mapsto \mathbb{R}$ denote the i-th model. The aggregated gradient g_{agg} has the following coordinate-wise value,

$$|g_{\text{agg}}(\mathbf{z})| = \text{coordinate-wise} \min_{i=1}^{N} |\nabla f_i(\mathbf{z})|\,.$$

The sign of g_{agg} at each coordinate is defined as the sign of the gradient f_i that achieves the minimimum absolute value. We remark that the aggregated gradient maintains the stable components in the gradients and discounts the unstable components. This is because, taking the minimum of gradient values ensures that the aggregated gradient drives the image generation if and only if all gradients from 18 models agree with each other.

D.2.6 Visualization of CNN filters

In this subsection, we provide visualization of CNN filters. These visualizations show that CNN features encode much richer patterns that Gabor wavelets do. They support

our finding that CNN based models perform better than simple Gabor wavelet based models in modelling V4 neurons. Because of the pooling operations, normalization operations, and non-linear activation functions in the CNNs, the CNN features are complex nonlinear functions of the raw input image. These CNN features are the outcome of learning from the large scale image dataset ImageNet, and are in general hard to explain via mathematical formula.

Inspired by the recent advances in CNN visualization [208, 206], we visualize the filters Layer 2, 3 and 4 of the AlexNet as follows: taking Layer 2 as an example, for each of the 256 types of filters, we exaustively search for nine image patches, from a dataset of one million image patches generated from ImageNet, that has the maximal output responses for the filter. The one million image patches are generated by randomly cropping images in ImageNet.

Figure D.3 shows a subset of 256 types of filters in Layer 2 of AlexNet. We have manually clustered these filters in categories. We observe that other than encoding edge-shape patterns, Layer 2 of AlexNet also encodes a rich set of curvature patterns, contour-blob patterns as well as crossing patterns. These patterns could be very useful in building a predictive model for V4 neurons, because similar shape tuning properties of V4 neurons have been reported before [169].

Similarly, Figure D.4 and Figure D.5 shows a subset filters in Layer 3 and Layer 4 of AlexNet. We observe that these filters encode even richer shape patterns. Some concrete patterns such as “dog head” and “birds” appear in Layer 3 filters. It has been shown that higher layers compute more global complex features than lower layers [208]. Unfortunately, the higher layer features become more specific to the classification task used to train AlexNet. It is not clear the higher layer features are as transferable as the lower layer features to other tasks [177].

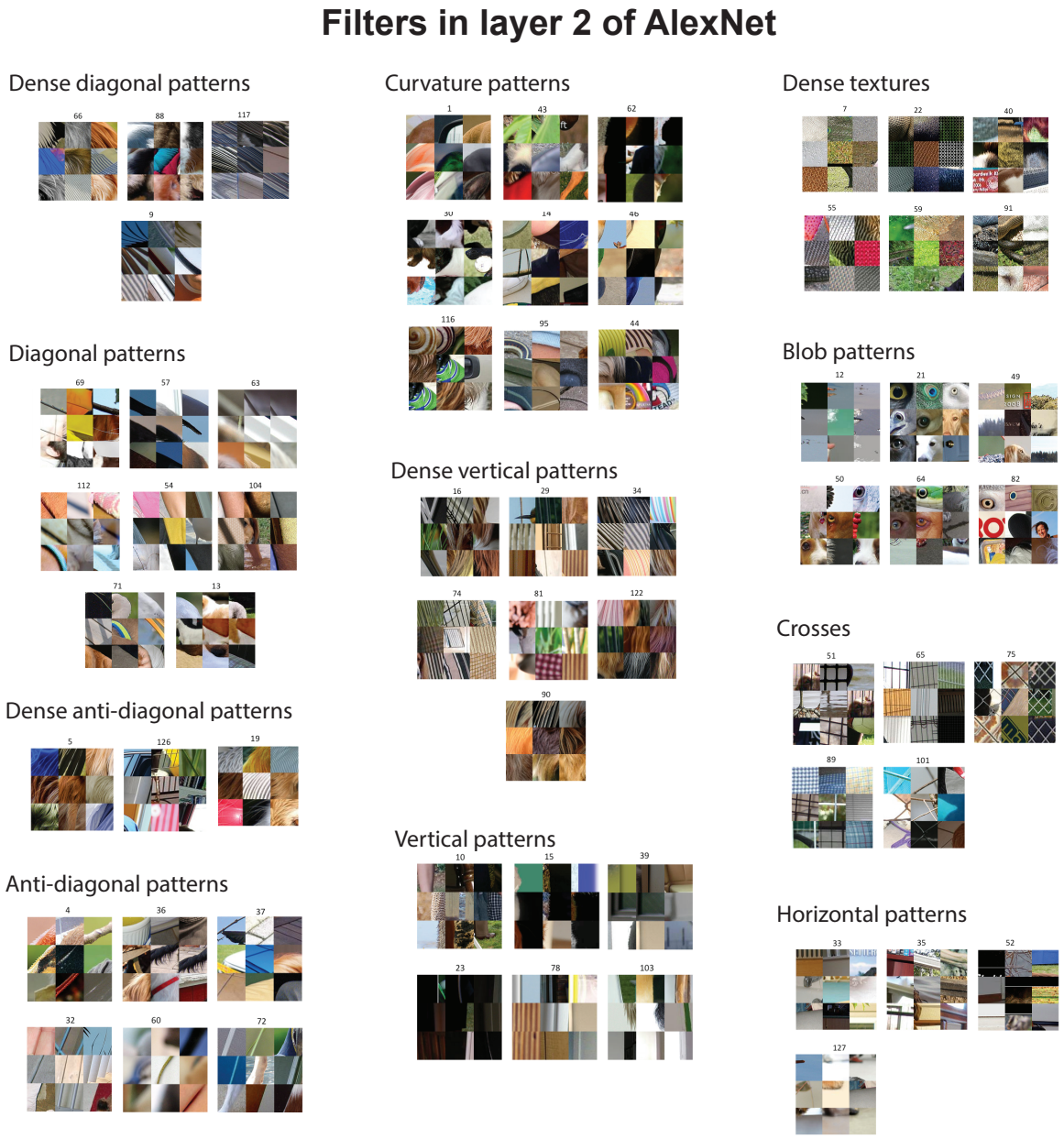


Figure D.3. Subset of filters in Layer 2 of AlexNet. To visualize each filter, we have fed one million image to the CNN and visualized top nine image patches that activate that has the maximal output responses for the filter [208]. We have manually clustered filters into categories.

Figure D.4. Subset of filters in Layer 3 of AlexNet. To visualize each filter, we have fed one million image to the CNN and visualized top nine image patches that activate that has the maximal output responses for the filter [208]. We have manually clustered filters into categories.

Figure D.5. Subset of filters in Layer 4 of AlexNet. To visualize each filter, we have fed one million image to the CNN and visualized top nine image patches that activate that has the maximal output responses for the filter [208]. We have manually clustered filters into categories.

D.3 Stability of analysis

In this section, we discuss the stability of our analysis for DeepTune images and model selected features.

D.3.1 Stability of DeepTune images

Our main analysis is based on DeepTune Images. The CNN-based approach for interpretation is potentially biased because of the specific choice of architecture, parametrization and methods. In this section, we investigate the convergence and stability of DeepTune visualization to different perturbations. Additionally, we study the stability of selected CNN features and weight-maps.

Convergence

To visualize the DeepTune image optimization process, we use SuperHeat visualization package to plot the heatmap of the CNN feature activation map throughout the optimization process in Figure D.6. There is a transition of the CNN feature activation map at about DeepTune optimization iteration 8. After this iteration, the CNN feature activation map stabilizes. The inactive columns correspond to the color-selective features in AlexNet. Our stimulus is gray-scale, therefore, it is expected to observe weak selection for these filters.

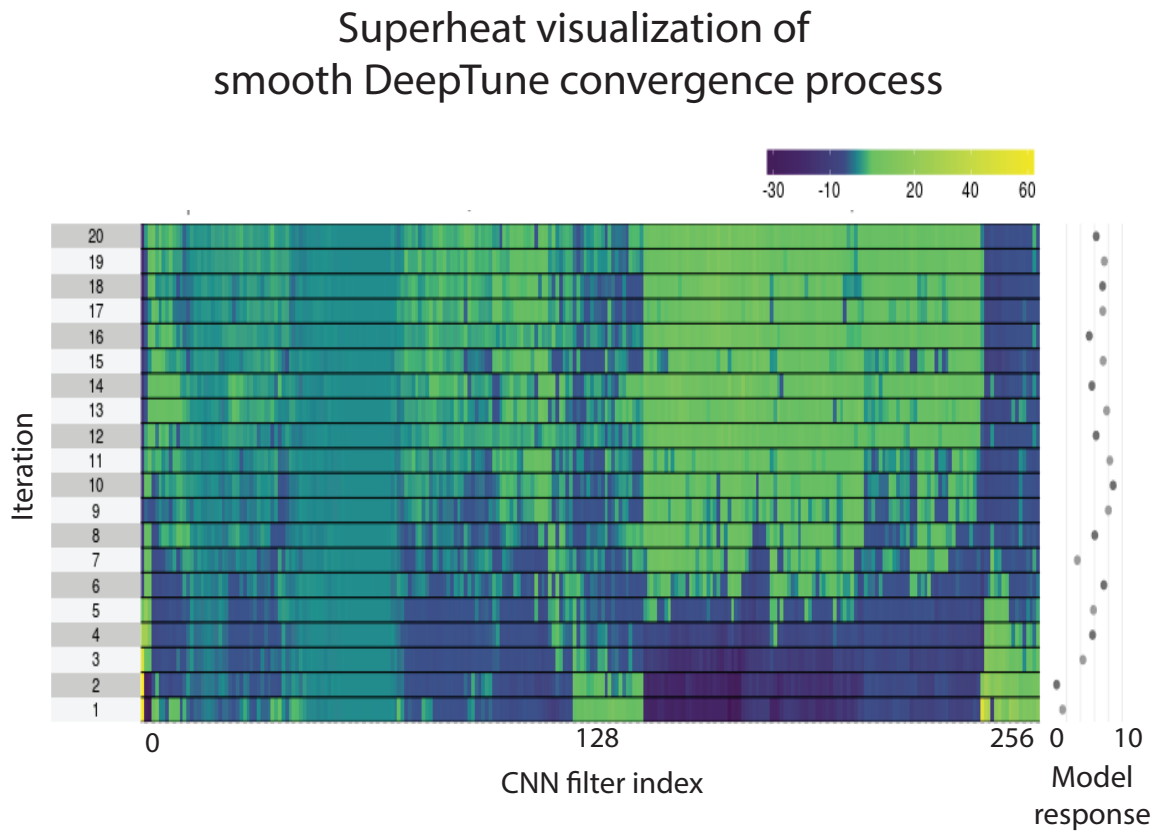


Figure D.6. Heatmap of the DeepTune image optimization process. We use SuperHeat visualization package to plot the heatmap of the CNN feature activation map throughout the optimization process.

Stability across different initialization

A DeepTune image is the final result of an optimization process on an initial random image. To study the effect of random initialization on the final DeepTune image, we run the optimization process on 10 different random starting image for each neuron. Figure D.7 shows 10 DeepTune images from these different initializations for five neurons. The patterns from 10 DeepTune images are visually similar. The average pair-wise correlation coefficient between 10 images is 0.97 for neuron 1. For other neurons, this value is not less than 0.94.

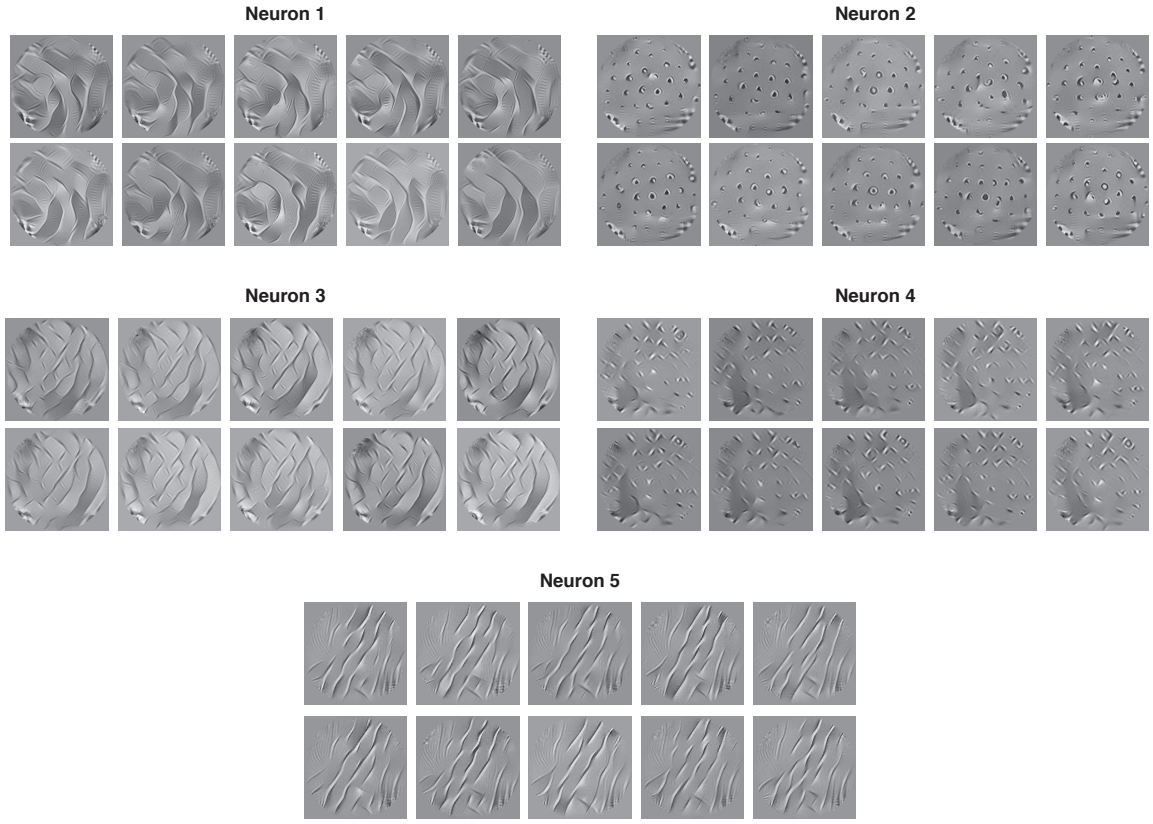


Figure D.7. DeepTune images with 10 different random initializations for five neurons.

Stability across 18 models

The DeepTune images from all of the 18 models studies in this chapter has stable patterns for each neuron. To construct the 18 models, we have used 3 pre-trained convolutional neural networks (AlexNet, VGG, and GoogleNet). From each network, we use either two, three, or four layers to extract features from images in neuroscience experiments. These features predict the spike rates of each neurons using a regularized linear regression. We use both l_2 (ridge regression) and l_1 (LASSO) regularizations. This results in 18 models for each neuron (3 networks, 3 layers, and 2 regression model). Figure D.8 shows DeepTune images from each of these 18 models for two neurons. The stable pattern among these 18 DeepTunes should be interpreted as the pattern that activates the neuron.

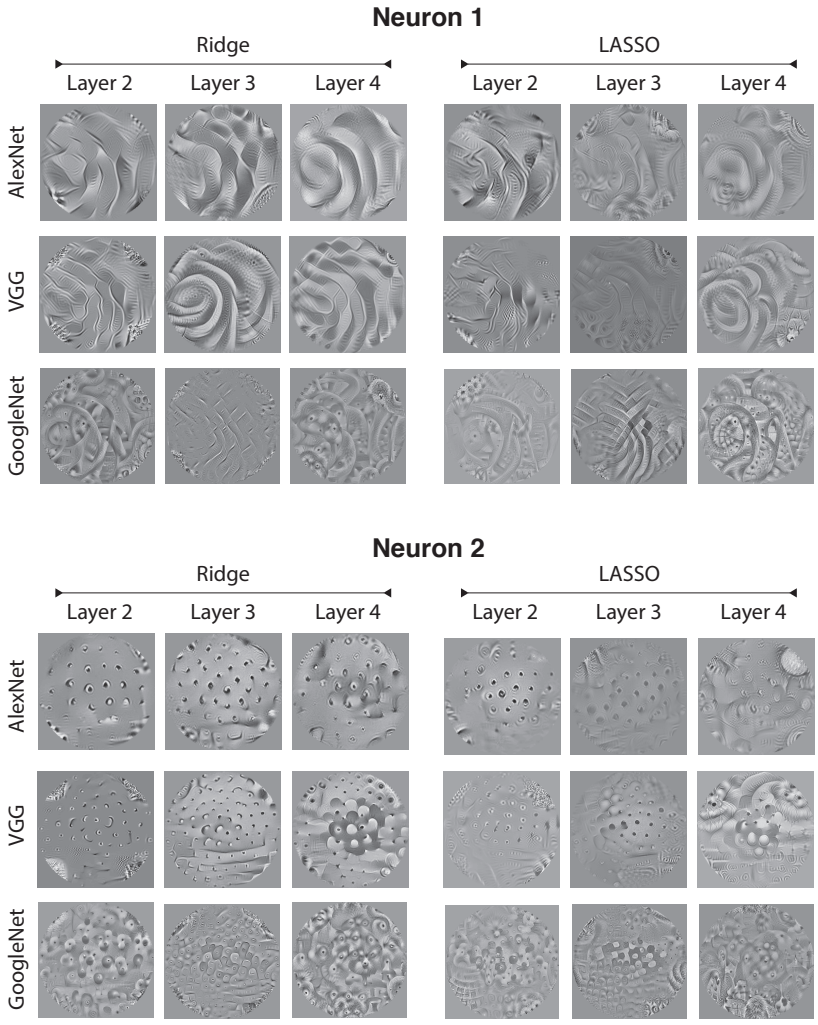


Figure D.8. Stability of the interpretable patterns in DeepTune images for neurons 1 and 2 across 18 models. The DeepTune images from Layer 4 + LASSO have artifacts that are not stable or consistent with the rest of DeepTune images. They should be discounted.

Stability of inhibitory DeepTune across models

In addition the the excitatory DeepTune images, Figure D.9 shows inhibitory DeepTune images from each of 9 models (3 networks, 3 layers, ridge regression) for two neurons. The stable pattern among these DeepTunes should be interpreted as the pattern that inhibits the neuron.

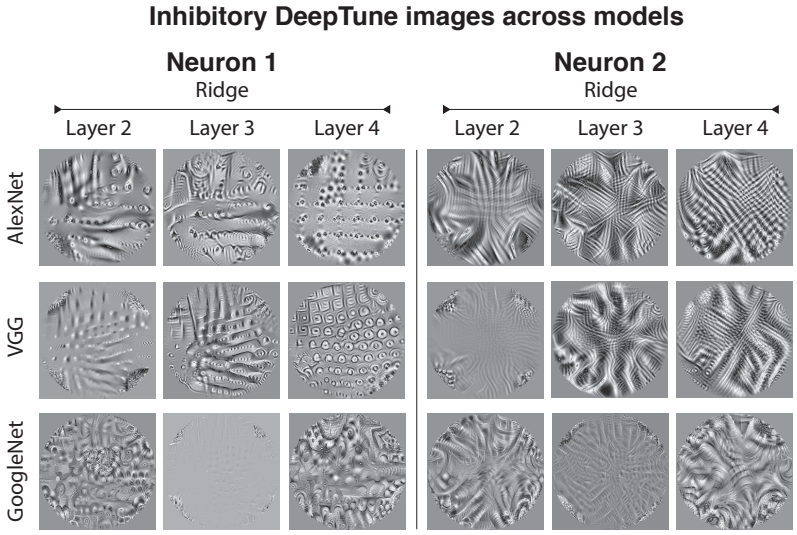


Figure D.9. Stability of the interpretable patterns in inhibitory DeepTune images for neurons 1 and 2 across 9 models.

Stability of identification matrix across models 1, 2, and 3

We first compute DeepTune images for each neuron and then construct a response identification matrix. For each DeepTune image image, we compute the responses from each of the 71 neuron models to it and plot them together in the identification matrix in the top heatmap plot in Figure D.10. The DeepTune image for each neuron has the highest response to model of that neuron compared to other neurons in the population (diagonal line visible in Figure D.10). No pairs of columns looks exactly identical which is an evidence that the 71 neurons' response properties are diverse. We also study the stability of this observation by feeding Deeptune images generated from VGG and GoogleNet models to AlexNet-based model. Figure D.10 the middle and bottom heatmap plots illustrates the responses of neuron models from AlexNet layer 2 to DeepTune images generated by VGG and GoogleNet layer 2 models. Ridge regression have been used in all of the models. The heatmaps in Figure D.10 contain clear diagonal patterns, showing the DeepTune images are stable across models. This observation, quantitatively confirms the visually observed stability of DeepTune images.

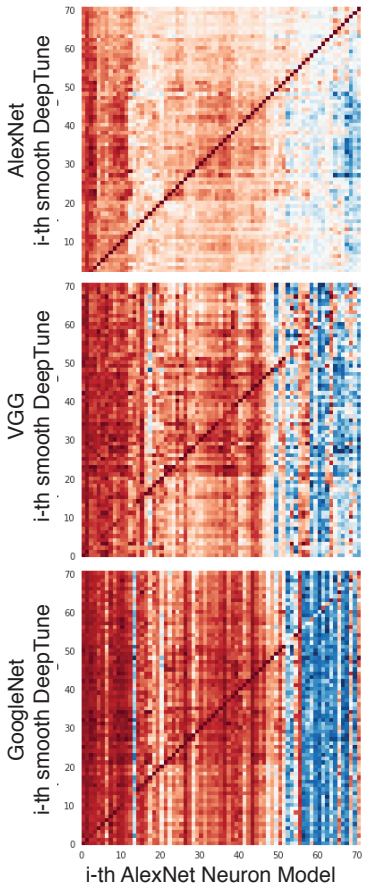


Figure D.10. DeepTune image identification matrix for three models. DeepTune images from layer 2 of AlexNet, VGG and GoogleNet are generated for each neuron. These images are fed into our prediction model based layer 2 feature of AlexNet. All three heatmaps contain clear diagonal pattern, showing the DeepTune images are stable across models. This observation, quantitatively confirms the visually observed stability of DeepTune images.

D.3.2 Stability of selected features and weight-Maps

In this section, we investigate the stability of CNN features selected by each neuron across different models. First, we visualize the top selected features and show that these features have stable visualization across models. Then, we use the regression coefficients in models to identify the model-inspired receptive field for each neuron. This is achieved by visualizing heatmaps of average regression coefficients across all features corresponding to each location in image.

Stability of top selected features across four main models for four neurons

Our model for each neuron consists of a CNN-based feature selection module and a linear regression model to predict the neuron spike rate from those features. Figure D.11 shows that these features have stable visualization across models. For neurons 2, 3, 4, and 5, we visualize the filters representing top two selected features. Each box with 9 image patches visualizes a filter in the CNN. To visualize the filter, we feed a million random natural images (from AlexNet dataset) to the network and show the top 9 image patches that activate the filter. For each model the left box corresponds to the top filter and the right box corresponds to the second top filter selected by neuron. The patterns are stable across all four models. For neuron 2, both top and second top filters for four models are selective to blob-like patterns. CNN filters selected by Neuron 3 like both curvatures and diagonal edges with 45 deg patterns. Neuron 3 prefers filters selective to corners and edges in both diagonals, however, no curvature filter is selected by this neuron. Neuron 5 is consistently selecting filters responsive to diagonal patterns in 45 deg. Similar observation holds for other V4 neurons on the population.

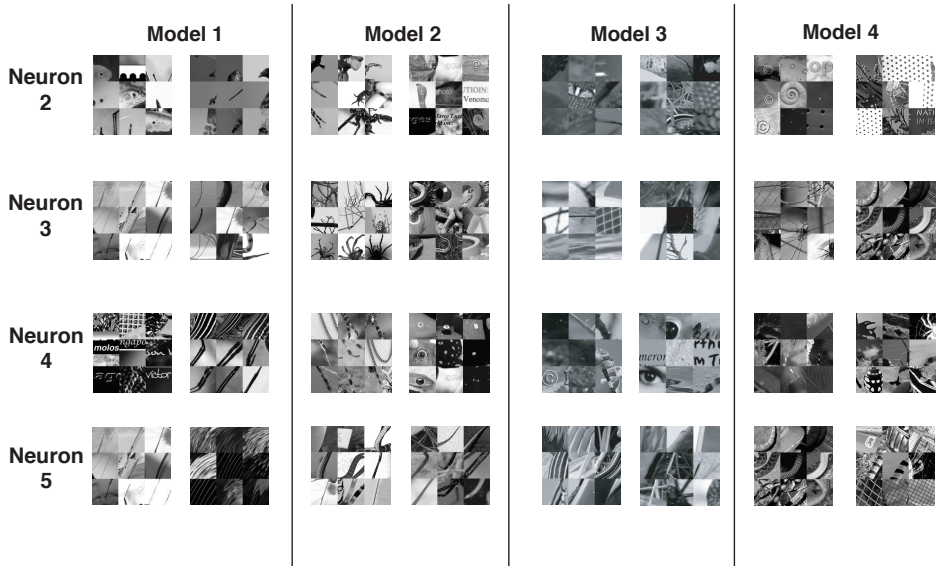


Figure D.11. Stability of top selected CNN features for each neuron across four main models. Each box visualizes a filter representing the feature in the CNN. To visualize the filter, we feed a million random natural images (from AlexNet dataset) to the network and show the top 9 image patches that activate the filter.

Stability of weight-maps across four main models

In this section, we study the stability of model-inspired spatial receptive fields for each neuron. The CNN features extracted from images have spatial structure due to nature of convolution operation. That is, each feature corresponds to a location on the image. Consequently, the regression coefficients mapping these features to neuron spike rates have similar spatial structure. After fitting the predictive models for each neuron, we estimate a model-inspired receptive field for each neuron, by averaging the regression coefficients in each location across different filters. The heatmap of average regression coefficients for each location on the image represents the importance of that location for the neuron. Figure D.12 shows these weight-maps for four neurons and four models. The weight-maps are stable across models. For neuron 2, the features in the center leaning to right side of the image are selected by the neuron. Neuron 3 and 5 are selective to features in a diagonal location. Neuron 4 prefers features in a cross-like location.

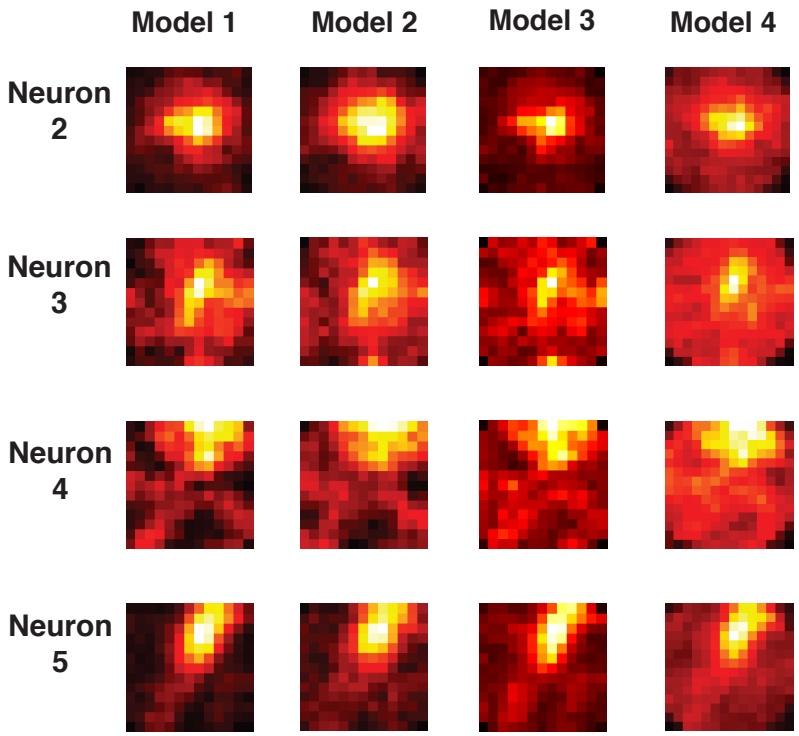


Figure D.12. Stability of average model weight-maps across four main models. These weight-maps estimates the model-inspired receptive filed. After fitting the predictive models for each neuron, we estimate a model-inspired receptive field for each neuron, by averaging the regression coefficients in each location across different filters. Each row corresponds to one neuron. Each column corresponds to a model. The weight-maps are stable across models for each neuron.

Stability LASSO vs Ridge

Inspecting raw coefficients from models learned by Lasso is problematic, however, due to the instability of the Lasso selected features. Particularly in cases where regressors are highly correlated (as is the case with features extracted from a CNNs), the model selection performed by Lasso may be inconsistent [209]. To overcome this issue and focus on the truly salient features for a particular neuron, we performed a stability analysis using 10-fold cross validation: the model was refit on each of the 10 perturbed datasets, and then the sets of selected variables were intersected. Model coefficients on this set were then averaged and used as a basis for our analysis. This is similar to the method introduced in Bach [8], except we use cross validation instead of bootstrap resampling.

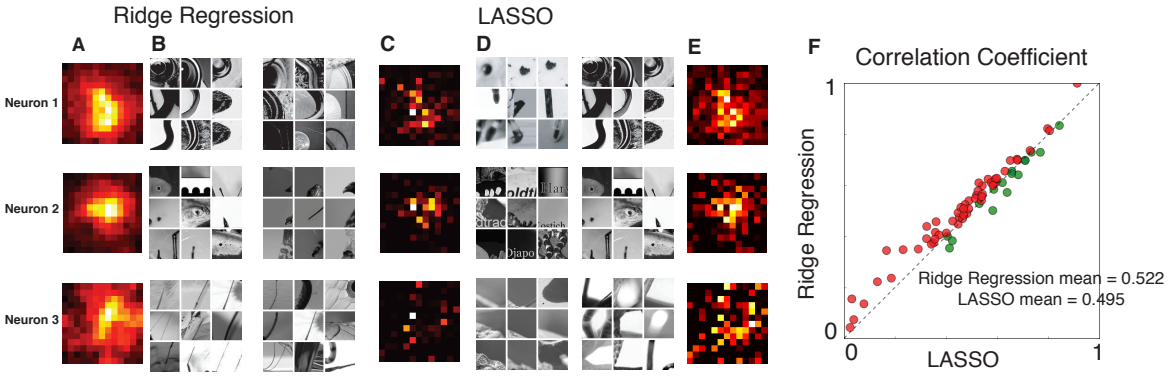


Figure D.13. Comparison of Lasso and Ridge feature selection. Ridge and Lasso give similar prediction performance. Lasso in general selects a smaller number of features (751 features in average) included in the set of features that Ridge selected (total 377,000 features). The top CNN filters that Lasso selected are similar to that of the Ridge regression.

D.4 Population analysis of V4 neurons

In this section, the excitatory and inhibitory DeepTune images are visualized for all of the 71 V4 neurons in the population.

D.4.1 Excitatory DeepTune images for all 71 V4 neurons

Figure D.14 shows the excitatory DeepTune images for all of the 71 neurons under study in visual area V4. The model used here is AlexNet layer 2 with ridge regression. Refer to the main text for a discussion on the diversity of the patterns excitatory by V4 neurons.

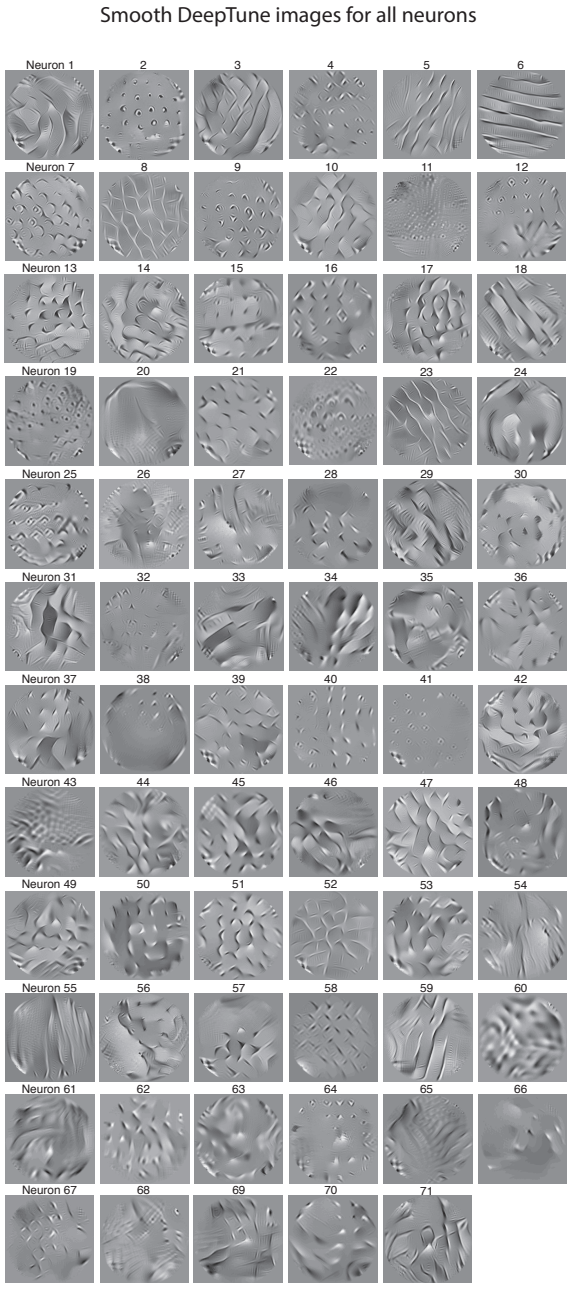


Figure D.14. DeepTune images for all 71 V4 neurons, based on AlexNet-Layer2 model

D.4.2 Inhibitory DeepTune images for all 71 V4 neurons

Figure D.15 illustrates the inhibitory DeepTune images for all of the 71 neurons. The model used here is AlexNet layer 2 with ridge regression. Most of the neurons have weak patterns in their inhibitory DeepTune image. For some of the neurons, the pattern is stronger. In the main text of chapter 3, we present a detailed discussion on the interpretation of patterns in inhibitory DeepTune images.

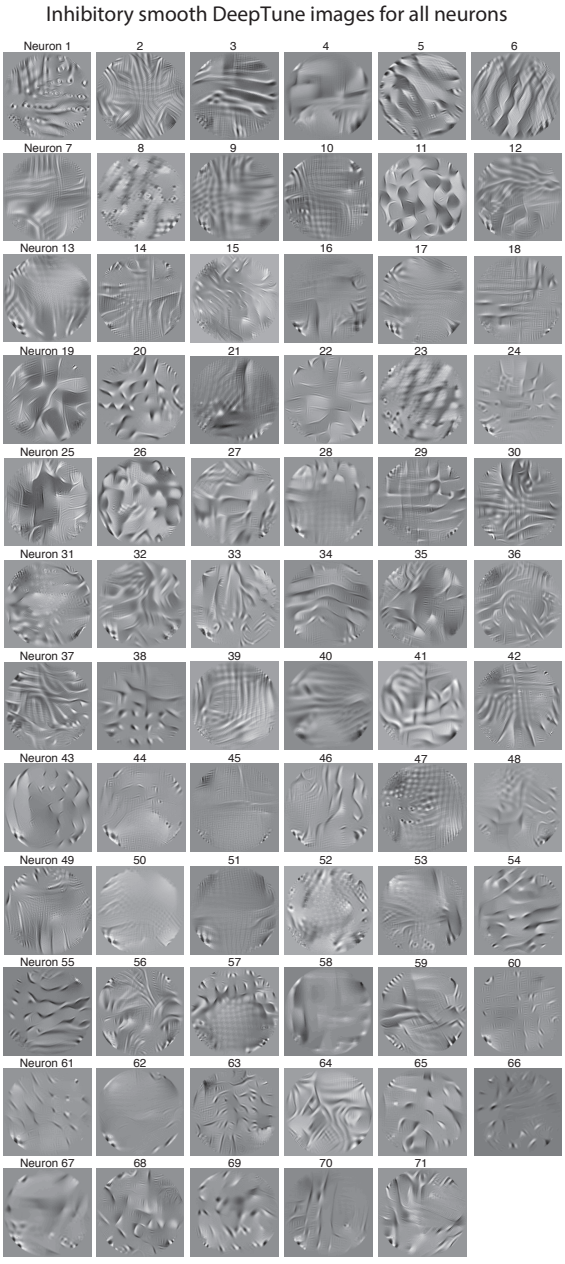


Figure D.15. Inhibitory DeepTune images for all 71 V4 neurons, based on AlexNet-Layer2 model

D.5 Analysis of our data based on previous methods

D.5.1 Spectral receptive field method (SRF)

It has been shown by David et al. [46] using spectral receptive field method (SRF) that many V4 neurons have more than one excitatory orientation tuning peak. Bimodal orientation tuning explains previous observations of selectivity for sharp corners [154]. Curvature or corner patterns will result in Bimodal orientation tuning in V4. We show via DeepTune that a large part of V4 neurons share this property and the result is consistent with that obtained by the spectral receptive field (SRF) [46].

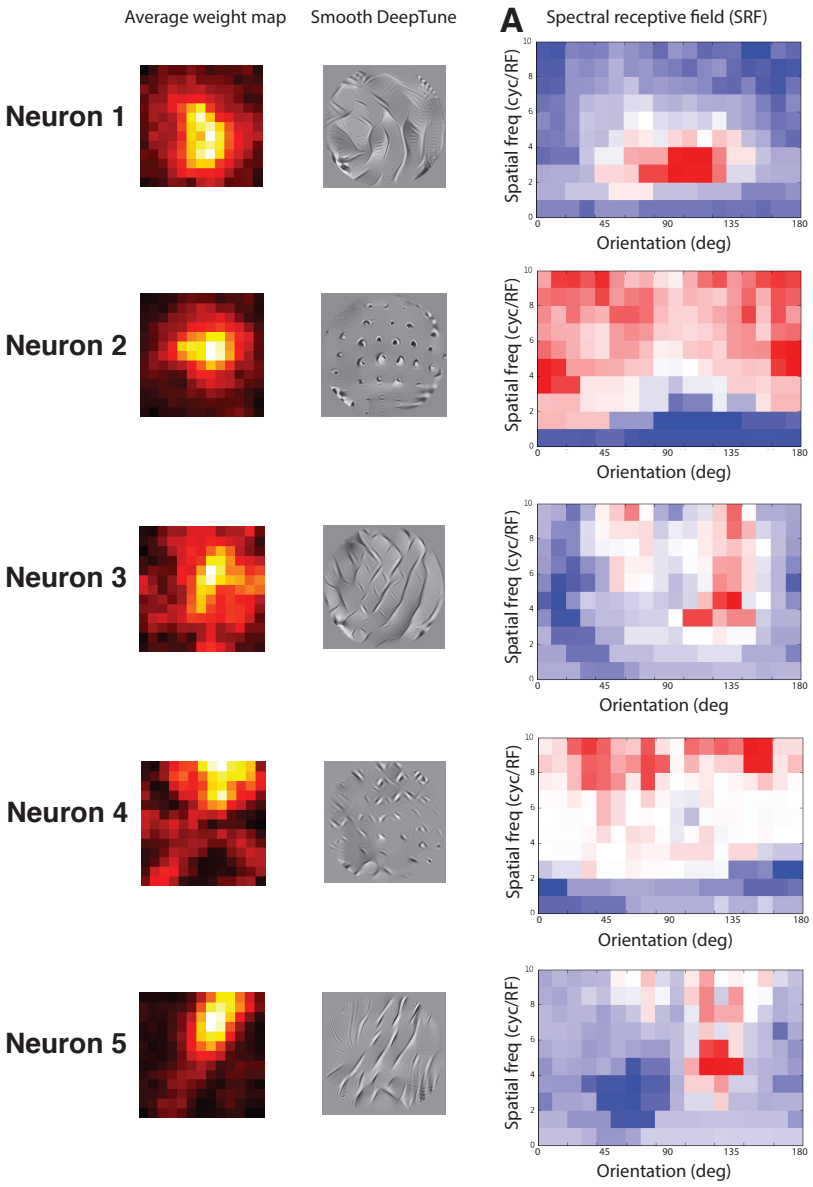


Figure D.16. Consistency of the average weight map and DeepTune images with spectral receptive field (SRF) [46].

D.6 Principal component analysis

Each V4 neuron model corresponds to a point in p-dimensional coefficient space; we can investigate the population of V4 neurons by examining their relative positions in this space. However, because p is very large (in the case of models based on N2 of AlexNet, $p = 389,376$) direct analysis of the coefficient vectors is impossible due to the curse of dimensionality. First, we perform ℓ^2 pooling of coefficient values across space and time delays to yield a single impact value for each of the filters in layer N2 of AlexNet. This gives a 256-dimensional representation, where each dimension corresponds to a single filter. Next, we perform principal components analysis (PCA) of the 71 points (each corresponding to a single V4 neuron) in this 256-dimensional space. PCA finds a set of linear transformations that capture a large proportion of the variance of the vectors. An examination of the coefficients of the loading vectors reveals that the first several principal components delineate several recognizable image features. The first principal component specifies whether neuron is selective to horizontal and vertical patterns. The second principal component delineates low-frequency patterns vs. dense blobs. The third principal component delineates diagonal vs non-diagonal smooth features.

Figure D.17.A shows the plot of the 71 V4 neurons according to their values in the first two principal components. Each neuron is shown via its DeepTune image. The color of DeepTune image borders is proportional to the third principle component with red being the highest PC value and blue the lowest. The neurons with highest values in PC 1 are selective to Figure D.17.B illustrates the 71 V4 neurons according to other principal components. The coefficients of the loading vectors for first three principal components are shown in Figure D.17.C. For the top coefficients, the corresponding filter is visualized by top 6 image patch that activate that filter. These image patches are found by feeding one million random image to the CNN and selecting the patches with highest filter response.

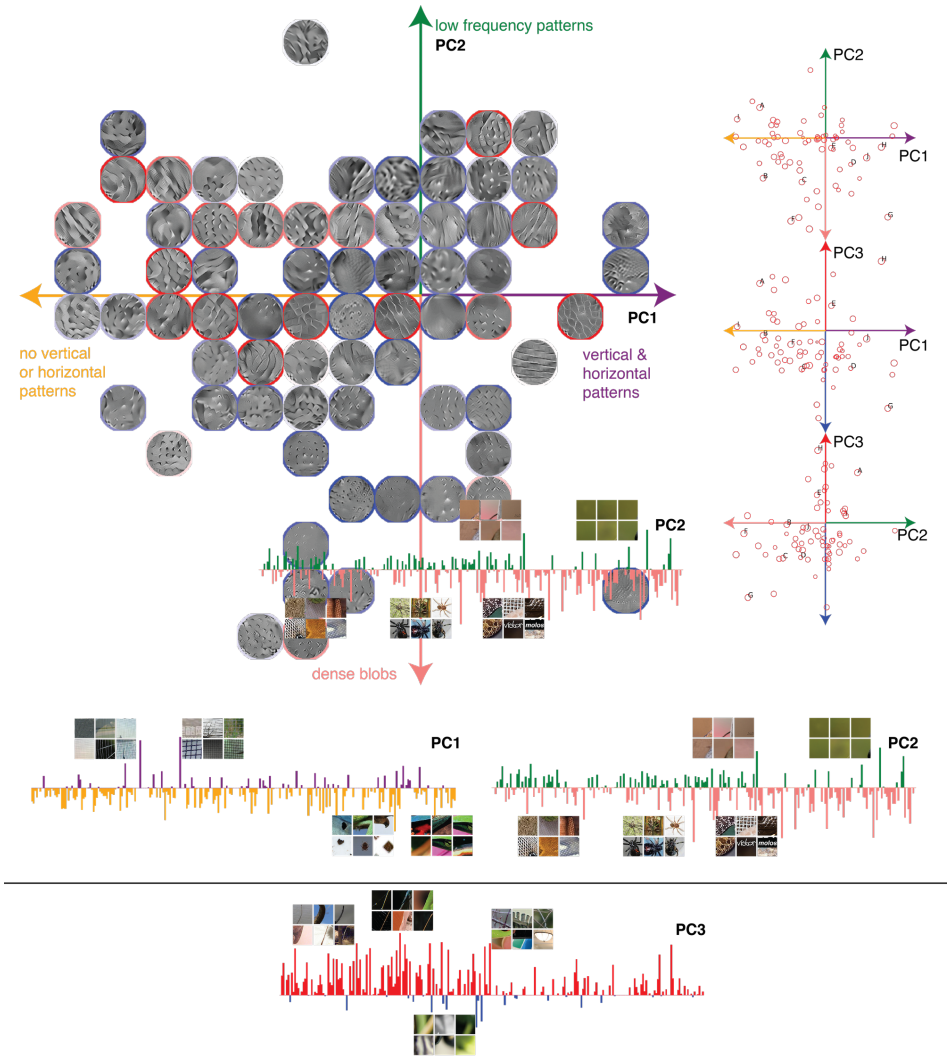


Figure D.17. Principal components analysis (PCA) of V4 neuron's population. **A.** 71 V4 neurons according to their values in the first two principal components. To compute the principle components, we perform ℓ^2 pooling of coefficient values across space and time delays to yield a single impact value for each of the filters in layer N2 of AlexNet. This gives a 256-dimensional representation, where each dimension corresponds to a single filter. Then, we perform principal components analysis (PCA) of the 71 points (each corresponding to a single V4 neuron) in this 256-dimensional space. Each neuron is shown via its DeepTune image. The color of DeepTune image borders is proportional to the third principle component with red being the highest PC value and blue the lowest. **B.** 71 V4 neurons according to other pairs of principal components. **C.** Coefficients of the loading vectors for the top three principal components. For the coefficients with highest values, the corresponding filter is visualized by top 6 image patch that activate that filter. These image patches are found by feeding one million random image to the CNN and selecting the patches with highest filter response.

D.7 Additional figures

D.7.1 Responses of our model to hand-crafted stimuli

In this section, we investigate the response of our CNN-based neuron models to polar, hyperbolic, and Cartesian gratings. We manually create images in each category based on the equations give in [64]. The response of the V4 models based on second layer of AlexNet with Ridge regression are computed for each of these hand crafted images. Figures D.18, D.19, and D.20 show the responses of three neurons to these images. The DeepTune image is also shown in each figure. The hand crafted images selected by the model are consistent with the patterns visible in the DeepTune image.

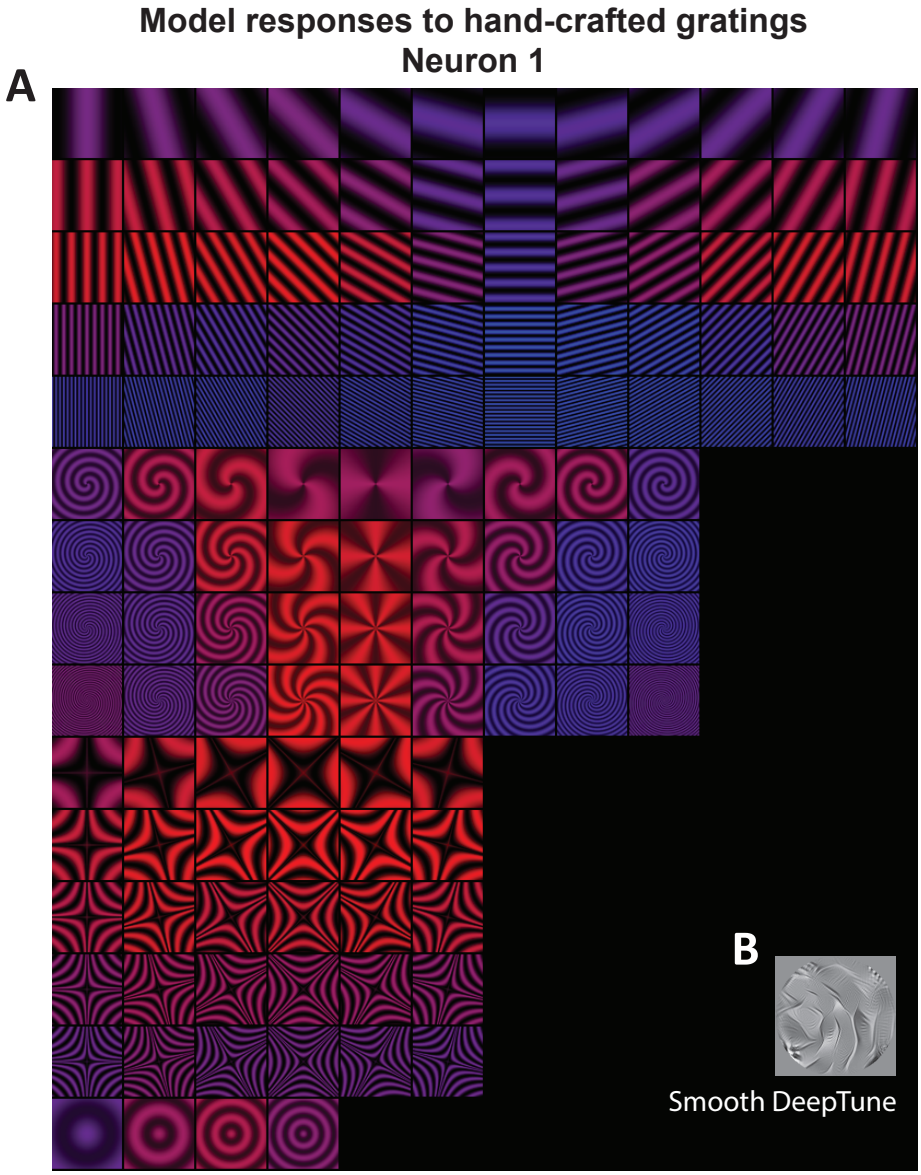


Figure D.18. Responses of AlexNet-Layer2 model to handcrafted stimuli for neuron 1. A. Responses of neuron 1 model to polar, hyperbolic, and Cartesian gratings. The gratings in red and blue correspond to excitatory and inhibitory stimulus, respectively. B. DeepTune of neuron 1

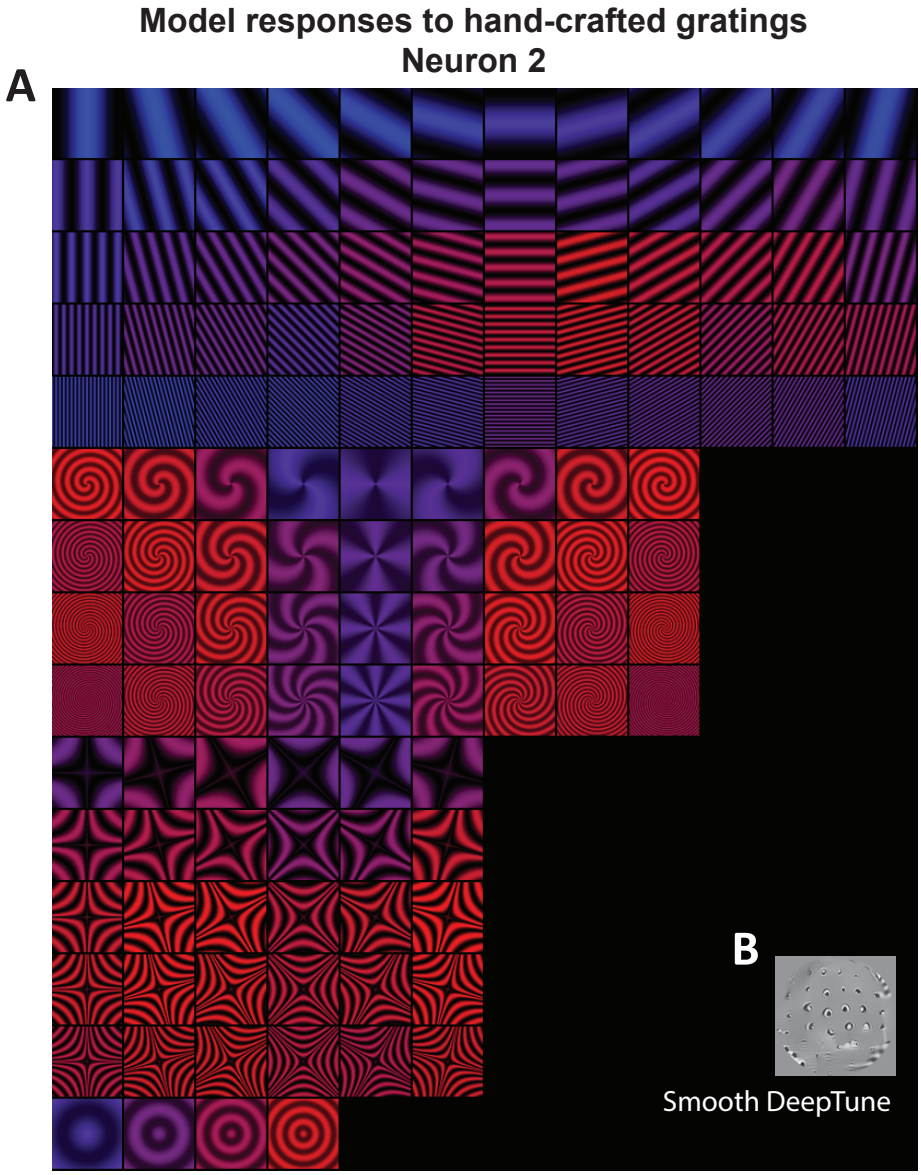


Figure D.19. Responses of AlexNet-Layer2 model to handcrafted stimuli for neuron 2. A. Responses of neuron 2 model to polar, hyperbolic, and Cartesian gratings. The gratings in red and blue correspond to excitatory and inhibitory stimulus, respectively. B. DeepTune of neuron 2

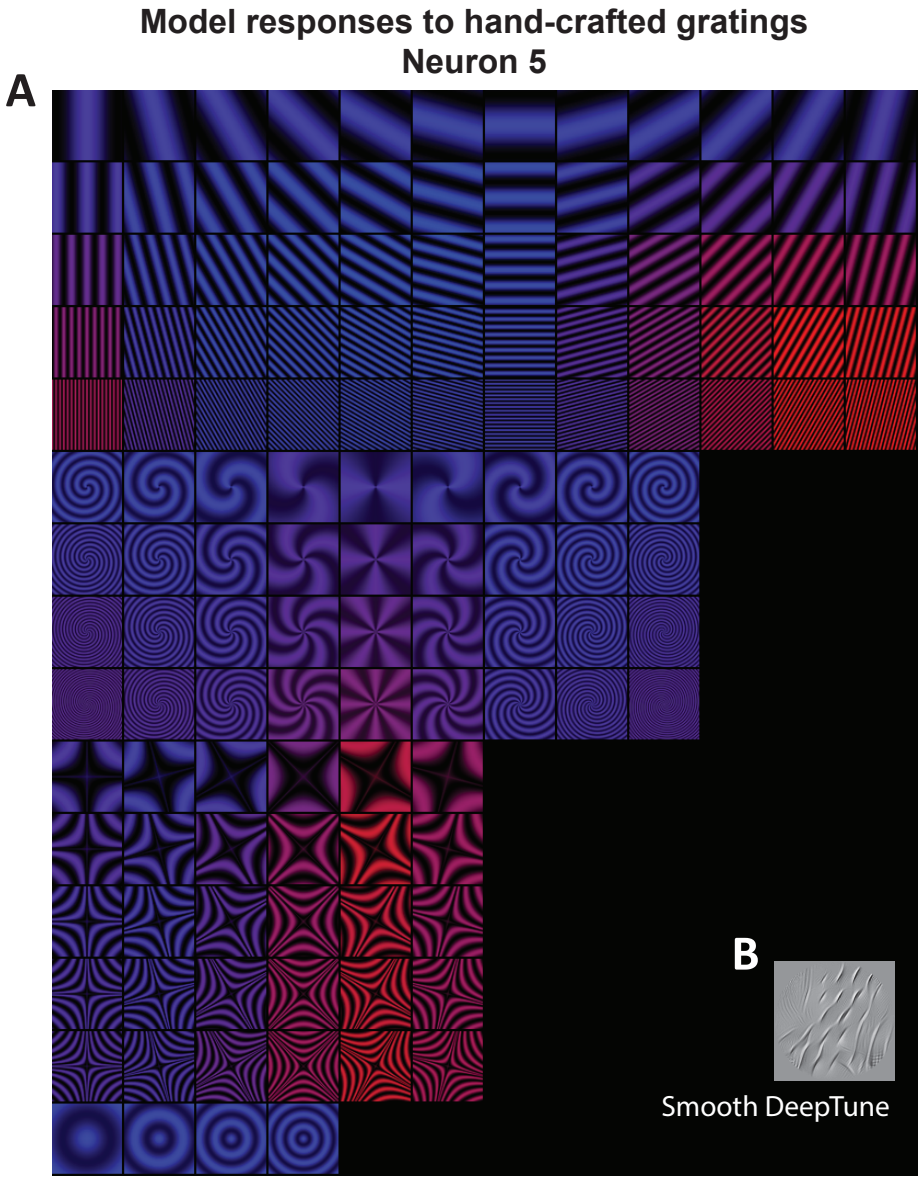


Figure D.20. Responses of AlexNet-Layer2 model to handcrafted stimuli for neuron 5. A. Responses of neuron 5 model to polar, hyperbolic, and Cartesian gratings. The gratings in red and blue correspond to excitatory and inhibitory stimulus, respectively. B. DeepTune of neuron 5

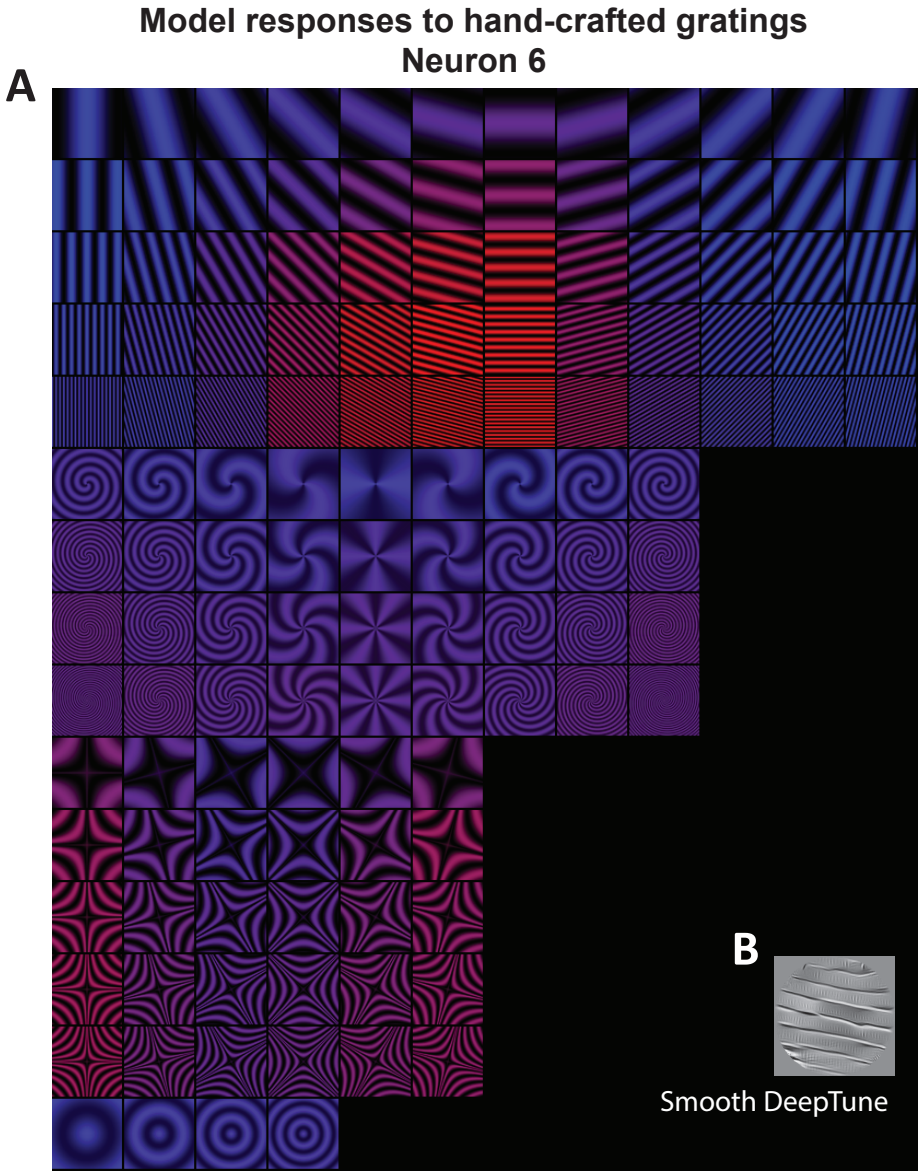


Figure D.21. Responses of AlexNet-Layer2 model to handcrafted stimuli for neuron 6. A. Responses of neuron 6 model to polar, hyperbolic, and Cartesian gratings. The gratings in red and blue correspond to excitatory and inhibitory stimulus, respectively. B. DeepTune of neuron 6

Bibliography

[1] Reza Abbasi-Asl et al. "The DeepTune framework for modeling and characterizing neurons in visual cortex area V4". In: *bioRxiv* (2018), p. 465534.

[2] Alekh Agarwal and Leon Bottou. "A lower bound for the optimization of finite sums". In: *International Conference on Machine Learning*. 2015, pp. 78–86.

[3] Berni J Alder and T E Wainwright. "Studies in molecular dynamics. I. General method". In: *The Journal of Chemical Physics* 31.2 (1959), pp. 459–466.

[4] David Aldous and James Allen Fill. *Reversible Markov chains and random walks on graphs*. Unfinished monograph, recompiled 2014, available at http://www.stat.berkeley.edu/~aldous/RWG/book.html. 2002.

[5] John Allman, Francis Miezin, and EveLynn McGuinness. "Stimulus specific responses from beyond the classical receptive field: neurophysiological mechanisms for local-global comparisons in visual neurons". In: *Annual review of neuroscience* 8.1 (1985), pp. 407–430.

[6] Kurt M Anstreicher. "The volumetric barrier for semidefinite programming". In: *Mathematics of Operations Research* 25.3 (2000), pp. 365–380.

[7] Fabrice Arcizet, Christophe Jouffrais, and Pascal Girard. "Natural textures classification in area V4 of the macaque monkey". In: *Experimental brain research* 189.1 (2008), pp. 109–120.

[8] Francis R Bach. "Bolasso: model consistent LASSO estimation through the bootstrap". In: *Proceedings of the 25th international conference on Machine learning*. ACM. 2008, pp. 33–40.

[9] Franck Barthe and Bernard Maurey. "Some remarks on isoperimetry of Gaussian type". In: *Annales de l'Institut Henri Poincare (B) Probability and Statistics*. Vol. 36. Elsevier. 2000, pp. 419–434.

[10] Peter L Bartlett, Michael I Jordan, and Jon D McAuliffe. "Convexity, classification, and risk bounds". In: *Journal of the American Statistical Association* 101.473 (2006), pp. 138–156.

[11] Peter L Bartlett and Shahar Mendelson. "Rademacher and Gaussian complexities: Risk bounds and structural results". In: *The Journal of Machine Learning Research* 3 (2003), pp. 463–482.

[12] Claude JP Bélisle, H Edwin Romeijn, and Robert L Smith. "Hit-and-run algorithms for generating multivariate distributions". In: *Mathematics of Operations Research* 18.2 (1993), pp. 255–266.

[13] Dimitris Bertsimas and Santosh Vempala. "Solving convex programs by random walks". In: *Journal of the ACM (JACM)* 51.4 (2004), pp. 540–556.

[14] MJ Betancourt, Simon Byrne, and Mark Girolami. "Optimizing the integrator step size for Hamiltonian Monte Carlo". In: *arXiv preprint arXiv:1411.6669* (2014).

[15] Rajendra Bhatia. *Matrix Analysis*. Vol. 169. Springer Science & Business Media, 2013.

[16] Sergey G Bobkov. "Isoperimetric and analytic inequalities for log-concave probability measures". In: *The Annals of Probability* 27.4 (1999), pp. 1903–1921.

[17] Nawaf Bou-Rabee, Andreas Eberle, and Raphael Zimmer. "Coupling and convergence for Hamiltonian Monte Carlo". In: *arXiv preprint arXiv:1805.00452* (2018).

[18] Nawaf Bou-Rabee and Martin Hairer. "Nonasymptotic mixing of the MALA algorithm". In: *IMA Journal of Numerical Analysis* 33.1 (2012), pp. 80–110.

[19] Olivier Bousquet and Léon Bottou. "The tradeoffs of large scale learning". In: *Advances in neural information processing systems*. 2008, pp. 161–168.

[20] Olivier Bousquet and André Elisseeff. "Stability and generalization". In: *The Journal of Machine Learning Research* 2 (2002), pp. 499–526.

[21] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[22] P. Brémaud. *Markov Chains, Gibbs Fields, Monte Carlo Simulation, and Queues*. Springer, 1991.

[23] Steve Brooks et al. *Handbook of Markov Chain Monte Carlo*. Chapman and Hall/CRC, 2011.

[24] Sébastien Bubeck et al. "Convex optimization: algorithms and complexity". In: *Foundations and Trends in Machine Learning* 8.3-4 (2015), pp. 231–357.

[25] Sébastien Bubeck, Ronen Eldan, and Joseph Lehec. "Sampling from a log-concave distribution with projected Langevin Monte Carlo". In: *arXiv preprint arXiv:1507.02564* (2015).

[26] Peter Bühlmann and Bin Yu. "Boosting with the L2 loss: regression and classification". In: *Journal of the American Statistical Association* 98.462 (2003), pp. 324–339.

[27] Peter J Bushell. "Hilbert's metric and positive contraction mappings in a Banach space". In: *Archive for Rational Mechanics and Analysis* 52.4 (1973), pp. 330–338.

[28] Charles F Cadieu et al. "Deep neural networks rival the representation of primate IT cortex for core visual object recognition". In: *PLoS Comput Biol* 10.12 (2014), e1003963.

[29] Matteo Carandini et al. "Do we know what the early visual system does?" In: *Journal of Neuroscience* 25.46 (2005), pp. 10577–10597.

[30] Eric T Carlson et al. "A sparse object coding scheme in area V4". In: *Current Biology* 21.4 (2011), pp. 288–293.

[31] Bob Carpenter et al. "Stan: a probabilistic programming language". In: *Journal of Statistical Software* 76.1 (2017).

[32] Zachary Charles and Dimitris Papailiopoulos. "Stability and generalization of learning algorithms that converge to global optima". In: *arXiv preprint arXiv:1710.08402* (2017).

[33] Jeff Cheeger. "A lower bound for the smallest eigenvalue of the Laplacian". In: *Proceedings of the Princeton Conference in honor of Professor S. Bochner*. 1969.

[34] Yuansi Chen, Chi Jin, and Bin Yu. "Stability and convergence trade-off of iterative optimization algorithms". In: *arXiv preprint arXiv:1804.01619* (2018).

[35] Yuansi Chen et al. "Fast MCMC sampling algorithms on polytopes". In: *The Journal of Machine Learning Research* 19.1 (2018), pp. 2146–2231.

[36] Yuansi Chen et al. "Fast mixing of Metropolized Hamiltonian Monte Carlo: Benefits of multi-step gradients". In: *arXiv preprint arXiv:1905.12247* (2019).

[37] Zongchen Chen and Santosh S Vempala. "Optimal convergence rate of Hamiltonian Monte Carlo for strongly logconcave distributions". In: *arXiv preprint arXiv:1905.02313* (2019).

[38] Xiang Cheng and Peter Bartlett. "Convergence of Langevin MCMC in KL-divergence". In: *arXiv preprint arXiv:1705.09048* (2017).

[39] Xiang Cheng et al. "Underdamped Langevin MCMC: A non-asymptotic analysis". In: *arXiv preprint arXiv:1707.03663* (2017).

[40] Ben Cousins and Santosh Vempala. "A cubic algorithm for computing Gaussian volume". In: *Proceedings of the twenty-fifth annual ACM-SIAM symposium on discrete algorithms*. Society for Industrial and Applied Mathematics. 2014, pp. 1215–1228.

[41] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. New York: John Wiley and Sons, 1991.

[42] Michael Creutz. "Global Monte Carlo algorithms for many-fermion systems". In: *Physical Review D* 38.4 (1988), p. 1228.

[43] Arnak S Dalalyan. "Theoretical guarantees for approximate sampling from smooth and log-concave densities". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* (2016).

[44] Arnak S Dalalyan and Avetik Karagulyan. "User-friendly guarantees for the Langevin Monte Carlo with inaccurate gradient". In: *Stochastic Processes and their Applications* (2019).

[45] John G Daugman. "Two-dimensional spectral analysis of cortical receptive field profiles". In: *Vision research* 20.10 (1980), pp. 847–856.

[46] Stephen V David, Benjamin Y Hayden, and Jack L Gallant. "Spectral receptive field properties explain shape selectivity in area V4". In: *Journal of neurophysiology* 96.6 (2006), pp. 3492–3505.

[47] Gregory C DeAngelis, Izumi Ohzawa, and RD Freeman. "Spatiotemporal organization of simple-cell receptive fields in the cat's striate cortex. II. Linearity of temporal and spatial summation". In: *Journal of Neurophysiology* 69.4 (1993), pp. 1118–1135.

[48] Bernard Delyon and Anatoli Juditsky. "Accelerated stochastic approximation". In: *SIAM Journal on Optimization* 3.4 (1993), pp. 868–881.

[49] Robert Desimone and Stanley J Schein. "Visual properties of neurons in area V4 of the macaque: sensitivity to stimulus form". In: *Journal of neurophysiology* 57.3 (1987), pp. 835–868.

[50] Luc P Devroye and Terry J Wagner. "Distribution-free performance bounds for potential function rules". In: *Information Theory, IEEE Transactions on* 25.5 (1979), pp. 601–604.

[51] Persi Diaconis and David Freedman. *On Markov chains with continuous state space*. Tech. rep. Technical Report, 1997.

[52] Persi Diaconis, Laurent Saloff-Coste, et al. "Logarithmic Sobolev inequalities for finite Markov chains". In: *The Annals of Applied Probability* 6.3 (1996), pp. 695–750.

[53] II Dikin. "Iterative solution to problems of linear and quadratic programming". In: *Doklady Akademii Nauk SSSR* 174.4 (1967), p. 747.

[54] Simon Duane et al. "Hybrid Monte Carlo". In: *Physics Letters B* 195.2 (1987), pp. 216–222.

[55] Alain Durmus and Eric Moulines. "High-dimensional Bayesian inference via the unadjusted Langevin algorithm". In: *arXiv preprint arXiv:1605.01559* (2016).

[56] Alain Durmus, Eric Moulines, and Marcelo Pereyra. "Efficient Bayesian computation by proximal Markov chain Monte Carlo: when Langevin meets Moreau". In: *arXiv preprint arXiv:1612.07471* (2016).

[57] Alain Durmus, Eric Moulines, and Eero Saksman. "On the convergence of Hamiltonian Monte Carlo". In: *arXiv preprint arXiv:1705.00166* (2017).

[58] Raaz Dwivedi et al. "Log-concave sampling: Metropolis-Hastings algorithms are fast". In: *arXiv preprint arXiv:1801.02309* (2018).

[59] Martin Dyer, Alan Frieze, and Ravi Kannan. "A random polynomial-time algorithm for approximating the volume of convex bodies". In: *Journal of the ACM (JACM)* 38.1 (1991), pp. 1–17.

[60] Andreas Eberle. "Error bounds for Metropolis-Hastings algorithms applied to perturbations of Gaussian measures in high dimensions". In: *The Annals of Applied Probability* 24.1 (2014), pp. 337–377.

[61] Bradley Efron. *The Jackknife, the Bootstrap, and Other Resampling Plans*. Vol. 38. Siam, 1982.

[62] J. Feldman, M. J. Wainwright, and D. R. Karger. "Using linear programming to decode binary linear codes". In: 51 (Mar. 2005), pp. 954–972.

[63] Alan Frieze, Ravi Kannan, and Nick Polson. "Sampling from log-concave distributions". In: *The Annals of Applied Probability* (1994), pp. 812–837.

[64] Jack L Gallant, Jochen Braun, and David C Van Essen. "Selectivity for polar, hyperbolic, and Cartesian gratings in macaque visual cortex". In: *Science* 259.5091 (1993), pp. 100–103.

[65] Jack L Gallant et al. "Neural responses to polar, hyperbolic, and Cartesian gratings in area V4 of the macaque monkey". In: *Journal of neurophysiology* 76.4 (1996), pp. 2718–2739.

[66] R Gattass, A P Sousa, and C G Gross. "Visuotopic organization and extent of V3 and V4 of the macaque". In: *The Journal of neuroscience* 8.6 (1988), pp. 1831–1845.

[67] Saul B Gelfand and Sanjoy K Mitter. "Recursive stochastic algorithms for global optimization in $\mathbb{R}^d$". In: *SIAM Journal on Control and Optimization* 29.5 (1991), pp. 999–1018.

[68] S. Geman and D. Geman. "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images". In: *IEEE Trans. PAMI* 6 (1984), pp. 721–741.

[69] Euhanna Ghadimi, Hamid Reza Feyzmahdavian, and Mikael Johansson. "Global convergence of the Heavy-ball method for convex optimization". In: *Control Conference (ECC), 2015 European*. IEEE. 2015, pp. 310–315.

[70] Walter R Gilks and Pascal Wild. "Adaptive rejection sampling for Gibbs sampling". In: *Applied Statistics* (1992), pp. 337–348.

[71] Sharad Goel, Ravi Montenegro, and Prasad Tetali. "Mixing time bounds via the spectral profile". In: *Electronic Journal of Probability* 11 (2006), pp. 1–26.

[72] Ulf Grenander and Michael I Miller. "Representations of knowledge in complex systems". In: *Journal of the Royal Statistical Society. Series B (Methodological)* (1994), pp. 549–603.

[73] Mikhail Gromov and Vitali D Milman. "A topological application of the isoperimetric inequality". In: *American Journal of Mathematics* 105.4 (1983), pp. 843–854.

[74] Adam Gustafson and Hariharan Narayanan. "John's walk". In: *arXiv preprint arXiv:1803.02032* (2018).

[75] Moritz Hardt, Ben Recht, and Yoram Singer. "Train faster, generalize better: Stability of stochastic gradient descent". In: *International Conference on Machine Learning*. 2016, pp. 1225–1234.

[76] Gilles Hargé. "A convex/log-concave correlation inequality for Gaussian measure and an application to abstract Wiener spaces". In: *Probability theory and related fields* 130.3 (2004), pp. 415–440.

[77] W Keith Hastings. "Monte Carlo sampling methods using Markov chains and their applications". In: *Biometrika* 57.1 (1970), pp. 97–109.

[78] Arthur E Hoerl and Robert W Kennard. "Ridge regression: biased estimation for nonorthogonal problems". In: *Technometrics* 12.1 (1970), pp. 55–67.

[79] Matthew D Hoffman and Andrew Gelman. "The No-U-Turn Sampler: adaptively setting path lengths in Hamiltonian Monte Carlo." In: *Journal of Machine Learning Research* 15.1 (2014), pp. 1593–1623.

[80] Roger A Horn and Charles R Johnson. *Matrix Analysis*. Cambridge University Press, 2012.

[81] Christian Houdré. "Mixed and isoperimetric estimates on the log-Sobolev constants of graphs and Markov chains". In: *Combinatorica* 21.4 (2001), pp. 489–513.

[82] Daniel Hsu, Sham Kakade, Tong Zhang, et al. "A tail inequality for quadratic forms of subgaussian random vectors". In: *Electronic Communications in Probability* 17 (2012).

[83] Kuo-Ling Huang and Sanjay Mehrotra. "An empirical evaluation of a walk-relax-round heuristic for mixed integer convex programs". In: *Computational Optimization and Applications* 60.3 (2015), pp. 559–585.

[84] Kuo-Ling Huang and Sanjay Mehrotra. "An empirical evaluation of walk-and-round heuristics for mixed integer linear programs". In: *Computational Optimization and Applications* 55.3 (2013), pp. 545–570.

[85] David H Hubel and Torsten N Wiesel. "Receptive fields and functional architecture of monkey striate cortex". In: *The Journal of physiology* 195.1 (1968), pp. 215–243.

[86] Leon Isserlis. "On a formula for the product-moment coefficient of any order of a normal frequency distribution in any number of variables". In: *Biometrika* 12.1/2 (1918), pp. 134–139.

[87] Svante Janson. *Gaussian Hilbert Spaces*. Vol. 129. Cambridge University Press, 1997.

[88] Søren Fiig Jarner and Ernst Hansen. "Geometric ergodicity of Metropolis algorithms". In: *Stochastic processes and their applications* 85.2 (2000), pp. 341–361.

[89] Mark Jerrum and Alistair Sinclair. "Conductance and the rapid mixing property for Markov chains: the approximation of permanent resolved". In: *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*. ACM. 1988, pp. 235–244.

[90] Yangqing Jia et al. "Caffe: Convolutional architecture for fast feature embedding". In: *Proceedings of the 22nd ACM international conference on Multimedia*. ACM. 2014, pp. 675–678.

[91] Chi Jin et al. "How to Escape Saddle Points Efficiently". In: *International Conference on Machine Learning*. 2017.

[92] Fritz John. *Extremum problems with inequalities as subsidiary conditions, Studies and Essays Presented to R. Courant on his 60th Birthday, January 8, 1948, 187–204*. 1948.

[93] Judson P Jones and Larry A Palmer. "An evaluation of the two-dimensional Gabor filter model of simple receptive fields in cat striate cortex". In: *Journal of neurophysiology* 58.6 (1987), pp. 1233–1258.

[94] Ravi Kannan, László Lovász, and Ravi Montenegro. "Blocking conductance and mixing in random walks". In: *Combinatorics, Probability and Computing* 15.4 (2006), pp. 541–570.

[95] Ravi Kannan, László Lovász, and Miklós Simonovits. "Isoperimetric problems for convex bodies and a localization lemma". In: *Discrete & Computational Geometry* 13.3-4 (1995), pp. 541–559.

[96] Ravi Kannan, László Lovász, and Miklós Simonovits. "Random walks and an o*(n5) volume algorithm for convex bodies". In: *Random Structures & Algorithms* 11.1 (1997), pp. 1–50.

[97] Ravindran Kannan and Hariharan Narayanan. "Random walks on polytopes and an affine interior point method for linear programming". In: *Mathematics of Operations Research* 37.1 (2012), pp. 1–20.

[98] S. C. Kapfer and W. Krauth. *Sampling from a polytope and hard-disk Monte Carlo*. Tech. rep. Ecole Normale Supérieure, 2013.

[99] Eucaly Kobatake and Keiji Tanaka. "Neuronal selectivities to complex object features in the ventral visual pathway of the macaque cerebral cortex". In: *Journal of neurophysiology* 71.3 (1994), pp. 856–867.

[100] Hideki Kondo and Hidehiko Komatsu. "Suppression on neuronal responses by a metacontrast masking stimulus in monkey V4". In: *Neuroscience research* 36.1 (2000), pp. 27–33.

[101] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[102] Samuel Kutin and Partha Niyogi. "Almost-everywhere algorithmic stability and generalization error". In: *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc. 2002, pp. 275–282.

[103] Jim Lawrence. "Polytope volume computation". In: *Mathematics of Computation* 57.195 (1991), pp. 259–271.

[104] Lucien Le Cam. "Asymptotic methods in statistical decision theory". In: (1986).

[105] B Boser Le Cun et al. "Handwritten digit recognition with a back-propagation network". In: *Advances in neural information processing systems*. Citeseer. 1990.

[106] Michel Ledoux. "Concentration of measure and logarithmic Sobolev inequalities". In: *Seminaire de Probabilites XXXIII*. Springer, 1999, pp. 120–216.

[107] Yin Tat Lee and Aaron Sidford. "Path finding methods for linear programming: Solving linear programs in $\tilde{O}(\sqrt{\text{rank}})$ iterations and faster algorithms for maximum flow". In: *Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium on*. IEEE. 2014, pp. 424–433.

[108] Yin Tat Lee, Zhao Song, and Santosh S Vempala. "Algorithmic Theory of ODEs and Sampling from Well-conditioned Logconcave Densities". In: *arXiv preprint arXiv:1812.06243* (2018).

[109] Yin Tat Lee and Santosh S Vempala. "Convergence rate of Riemannian Hamiltonian Monte Carlo and faster polytope volume computation". In: *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*. ACM. 2018, pp. 1115–1121.

[110] Yin Tat Lee and Santosh S Vempala. "Geodesic walks in polytopes". In: *arXiv preprint arXiv:1606.04696* (2016).

[111] Yin Tat Lee and Santosh S Vempala. "Stochastic localization+ Stieltjes barrier= tight bound for log-Sobolev". In: *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*. ACM. 2018, pp. 1122–1129.

[112] Yin Tat Lee and Santosh S. Vempala. "Convergence rate of riemannian Hamiltonian Monte Carlo and faster polytope volume computation". In: *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*. 2018, pp. 1115–1121.

[113] Yin Tat Lee and Santosh Srinivas Vempala. "Eldan's Stochastic Localization and the KLS Hyperplane Conjecture: An Improved Lower Bound for Expansion". In: *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE. 2017, pp. 998–1007.

[114] Samuel Livingstone et al. "On the geometric ergodicity of Hamiltonian Monte Carlo". In: *arXiv preprint arXiv:1601.08057* (2016).

[115] László Lovász. "Hit-and-run mixes fast". In: *Mathematical Programming* 86.3 (1999), pp. 443–461.

[116] László Lovász et al. "Random walks on graphs: A survey". In: *Combinatorics, Paul erdos is eighty* 2.1 (1993), pp. 1–46.

[117] László Lovász and Ravi Kannan. "Faster mixing via average conductance". In: *Proceedings of the 31st annual ACM Symposium on Theory of Computing*. ACM. 1999, pp. 282–287.

[118] László Lovász and Miklós Simonovits. "Random walks in a convex body and an improved volume algorithm". In: *Random Structures & Algorithms* 4.4 (1993), pp. 359–412.

[119] László Lovász and Miklós Simonovits. "The mixing rate of Markov chains, an isoperimetric inequality, and computing the volume". In: *Proceedings of 31st Annual Symposium on Foundations of Computer Science, 1990*. IEEE. 1990, pp. 346–354.

[120] László Lovász and Santosh Vempala. "Hit-and-run from a corner". In: *SIAM Journal on Computing* 35.4 (2006), pp. 985–1005.

[121] László Lovász and Santosh Vempala. "Hit-and-run is fast and fun". In: *Tehnical Report, Microsoft Research* (2003).

[122] László Lovász and Santosh Vempala. "Simulated annealing in convex bodies and an $O^*(n^4)$ volume algorithm". In: *Journal of Computer and System Sciences* 72.2 (2006), pp. 392–417.

[123] László Lovász and Santosh Vempala. "The geometry of logconcave functions and sampling algorithms". In: *Random Structures & Algorithms* 30.3 (2007), pp. 307–358.

[124] Yi-An Ma et al. "Sampling can be faster than optimization". In: *arXiv preprint arXiv:1811.08413* (2018).

[125] Aravindh Mahendran and Andrea Vedaldi. "Understanding deep image representations by inverting them". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 5188–5196.

[126] Michael W Mahoney et al. "Randomized algorithms for matrices and data". In: *Foundations and Trends in Machine Learning* 3.2 (2011), pp. 123–224.

[127] Oren Mangoubi and Aaron Smith. "Rapid Mixing of Hamiltonian Monte Carlo on Strongly Log-Concave Distributions". In: *arXiv preprint arXiv:1708.07114* (2017).

[128] Oren Mangoubi and Nisheeth K Vishnoi. "Dimensionally Tight Running Time Bounds for Second-Order Hamiltonian Monte Carlo". In: *arXiv preprint arXiv: 1802.08898* (2018).

[129] Oren Mangoubi and Nisheeth K Vishnoi. "Nonconvex sampling with the Metropolis-adjusted Langevin algorithm". In: *arXiv preprint arXiv:1902.08452* (2019).

[130] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: http://tensorflow.org/.

[131] John C Mason and David C Handscomb. *Chebyshev polynomials*. CRC Press, 2002.

[132] Kerrie L Mengersen, Richard L Tweedie, et al. "Rates of convergence of the Hastings and Metropolis algorithms". In: *The Annals of Statistics* 24.1 (1996), pp. 101–121.

[133] William H Merigan. "Cortical area V4 is critical for certain texture discriminations, but this effect is not dependent on attention". In: *Visual neuroscience* 17.6 (2000), pp. 949–958.

[134] Nicholas Metropolis et al. "Equation of state calculations by fast computing machines". In: *The Journal of Chemical Physics* 21.6 (1953), pp. 1087–1092.

[135] Sean P Meyn and Richard L Tweedie. *Markov Chains and Stochastic Stability*. Springer Science & Business Media, 2012.

[136] Sean P Meyn and Robert L Tweedie. "Computable bounds for geometric convergence rates of Markov chains". In: *The Annals of Applied Probability* (1994), pp. 981–1011.

[137] Ben Morris and Yuval Peres. "Evolving sets, mixing and heat kernel bounds". In: *Probability Theory and Related Fields* 133.2 (2005), pp. 245–266.

[138] Wenlong Mou et al. "Generalization Bounds of SGLD for Non-convex Learning: Two Theoretical Viewpoints". In: *arXiv preprint arXiv:1707.05947* (2017).

[139] Eric Moulines and Francis R Bach. "Non-asymptotic analysis of stochastic approximation algorithms for machine learning". In: *Advances in Neural Information Processing Systems*. 2011, pp. 451–459.

[140] Anirvan S Nandy et al. "The fine structure of shape tuning in area V4". In: *Neuron* 78.6 (2013), pp. 1102–1115.

[141] Hariharan Narayanan. "Randomized interior point methods for sampling and optimization". In: *The Annals of Applied Probability* 26.1 (2016), pp. 597–641.

[142] Hariharan Narayanan and Alexander Rakhlin. "Efficient sampling from time-varying log-concave distributions". In: *arXiv preprint arXiv:1309.5977* (2013).

[143] Thomas Naselaris et al. "Bayesian reconstruction of natural images from human brain activity". In: *Neuron* 63.6 (2009), pp. 902–915.

[144] Radford M Neal. "An improved acceptance procedure for the hybrid Monte Carlo algorithm". In: *Journal of Computational Physics* 111 (1994), pp. 194–203.

[145] Radford M Neal. "MCMC using Hamiltonian dynamics". In: *Handbook of Markov Chain Monte Carlo* 2.11 (2011).

[146] Arkadi Nemirovski et al. "Robust stochastic approximation approach to stochastic programming". In: *SIAM Journal on Optimization* 19.4 (2009), pp. 1574–1609.

[147] A-S Nemirovsky, D-B Yudin, and E-R Dawson. "Problem complexity and method efficiency in optimization." In: (1982).

[148] Yurii Nesterov. "A method of solving a convex programming problem with convergence rate $\mathcal{O}(1/k^2)$". In: *Soviet Mathematics Doklady*. Vol. 27. 2. 1983, pp. 372–376.

[149] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*. Vol. 87. Springer Science & Business Media, 2013.

[150] Yurii Nesterov and Arkadii Nemirovskii. *Interior-point polynomial algorithms in convex programming*. SIAM, 1994.

[151] Gouki Okazawa, Satohiro Tajima, and Hidehiko Komatsu. "Image statistics underlying natural texture selectivity of neurons in macaque V4". In: *Proceedings of the National Academy of Sciences* 112.4 (2015), E351–E360.

[152] G Parisi. "Correlation functions and computer simulations". In: *Nuclear Physics B* 180.3 (1981), pp. 378–384.

[153] Anitha Pasupathy and Charles E Connor. "Population coding of shape in area V4". In: *Nature neuroscience* 5.12 (2002), p. 1332.

[154] Anitha Pasupathy and Charles E Connor. "Responses to contour features in macaque area V4". In: *Journal of Neurophysiology* 82.5 (1999), pp. 2490–2502.

[155] Marcelo Pereyra. "Proximal Markov chain Monte Carlo algorithms". In: *Statistics and Computing* 26.4 (2016), pp. 745–760.

[156] Natesh S Pillai, Andrew M Stuart, Alexandre H Thiéry, et al. "Optimal scaling and diffusion limits for the Langevin algorithm in high dimensions". In: *The Annals of Applied Probability* 22.6 (2012), pp. 2320–2356.

[157] Boris T Polyak. "Some methods of speeding up the convergence of iteration methods". In: *USSR Computational Mathematics and Mathematical Physics* 4.5 (1964), pp. 1–17.

[158] Maxim Raginsky, Alexander Rakhlin, and Matus Telgarsky. "Non-convex learning via stochastic gradient Langevin dynamics: a nonasymptotic analysis". In: *arXiv preprint arXiv:1702.03849* (2017).

[159] Brian D Ripley. "Stochastic simulation. Vol. 316". In: *Manhattan: John Wiley and Sons* (2009), pp. 96–118.

[160] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer texts in statistics. New York, NY: Springer-Verlag, 1999.

[161] Christian P Robert. *Monte Carlo methods*. Wiley Online Library, 2004.

[162] Gareth O Roberts and Jeffrey S Rosenthal. "Complexity bounds for MCMC via diffusion limits". In: *arXiv preprint arXiv:1411.0712* (2014).

[163] Gareth O Roberts and Jeffrey S Rosenthal. "Optimal scaling for various Metropolis-Hastings algorithms". In: *Statistical Science* 16.4 (2001), pp. 351–367.

[164] Gareth O Roberts, Jeffrey S Rosenthal, et al. "General state space Markov chains and MCMC algorithms". In: *Probability Surveys* 1 (2004), pp. 20–71.

[165] Gareth O Roberts and Osnat Stramer. "Langevin diffusions and Metropolis-Hastings algorithms". In: *Methodology and computing in applied probability* 4.4 (2002), pp. 337–357.

[166] Gareth O Roberts and Richard L Tweedie. "Exponential convergence of Langevin distributions and their discrete approximations". In: *Bernoulli* (1996), pp. 341–363.

[167] Gareth O Roberts and Richard L Tweedie. "Geometric convergence and central limit theorems for multidimensional Hastings and Metropolis algorithms". In: *Biometrika* 83.1 (1996), pp. 95–110.

[168] J Cooper Roddey, B Girish, and John P Miller. "Assessing the performance of neural encoding models in the presence of noise". In: *Journal of computational neuroscience* 8.2 (2000), pp. 95–112.

[169] Anna W Roe et al. "Toward a unified theory of visual area V4". In: *Neuron* 74.1 (2012), pp. 12–29.

[170] William H Rogers and Terry J Wagner. "A finite sample distribution-free performance bound for local discrimination rules". In: *The Annals of Statistics* (1978), pp. 506–514.

[171] Leonid I Rudin, Stanley Osher, and Emad Fatemi. "Nonlinear total variation based noise removal algorithms". In: *Physica D: Nonlinear Phenomena* 60.1-4 (1992), pp. 259–268.

[172] Olga Russakovsky et al. "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision (IJCV)* 115.3 (2015), pp. 211–252. DOI: 10.1007/s11263-015-0816-y.

[173] Sushant Sachdeva and Nisheeth K Vishnoi. "The mixing time of the Dikin walk in a polytope—A simple proof". In: *Operations Research Letters* 44.5 (2016), pp. 630–634.

[174] Stanley J Schein and Robert Desimone. "Spectral properties of V4 neurons in the macaque". In: *Journal of Neuroscience* 10.10 (1990), pp. 3369–3389.

[175] Matthew T Schmolesky et al. "Signal timing across the macaque visual system". In: *Journal of neurophysiology* 79.6 (1998), pp. 3272–3278.

[176] Oliver Schoppe et al. "Measuring the performance of neural models". In: *Frontiers in computational neuroscience* 10 (2016).

[177] Ali Sharif Razavian et al. "CNN features off-the-shelf: an astounding baseline for recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2014, pp. 806–813.

[178] Eero P Simoncelli and Bruno A Olshausen. "Natural image statistics and neural representation". In: *Annual review of neuroscience* 24.1 (2001), pp. 1193–1216.

[179] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (2014).

[180] Adrian FM Smith and Gareth O Roberts. "Bayesian computation via the Gibbs sampler and related Markov chain Monte Carlo methods". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 55.1 (1993), pp. 3–23.

[181] BJ Smith. *Mamba: Markov chain Monte Carlo (MCMC) for Bayesian analysis in julia*. Software available at mambajl.readthedocs.io. 2014. URL: https://mambajl.readthedocs.io/en/latest/.

[182] Robert L Smith. "Efficient Monte Carlo procedures for generating points uniformly distributed over bounded regions". In: *Operations Research* 32.6 (1984), pp. 1296–1308.

[183] Christian Szegedy et al. "Going deeper with convolutions". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 1–9.

[184] Denis Talay and Luciano Tubaro. "Expansion of the global error for numerical schemes solving stochastic differential equations". In: *Stochastic analysis and applications* 8.4 (1990), pp. 483–509.

[185] Pafnutï Lvovitch Tchebychev. *Théorie des mécanismes connus sous le nom de parallélogrammes*. Imprimerie de l'Académie impériale des sciences, 1853.

[186] Robert Tibshirani. "Regression Selection and Shrinkage via the Lasso". In: *Journal of the Royal Statistical Society B* 58 (1994), pp. 267–288. ISSN: 00359246. DOI: 10.2307/2346178. URL: http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.35.7574.

[187] Robert Tibshirani. "Regression shrinkage and selection via the lasso". In: *Journal of the Royal Statistical Society. Series B (Methodological)* (1996), pp. 267–288.

[188] Luke Tierney. "Markov chains for exploring posterior distributions". In: *The Annals of Statistics* (1994), pp. 1701–1728.

[189] Jon Touryan and James A Mazer. "Linear and non-linear properties of feature selectivity in V4 neurons". In: *Frontiers in systems neuroscience* 9 (2015).

[190] Pravin M Vaidya. "A new algorithm for minimizing convex functions over convex sets". In: *30th Annual Symposium on Foundations of Computer Science, 1989*. IEEE. 1989, pp. 338–343.

[191] Pravin M Vaidya and David S Atkinson. "A technique for bounding the number of iterations in path following algorithms". In: *Complexity in Numerical Optimization*. World Scientific, 1993, pp. 462–489.

[192] Vladimir Vapnik, Esther Levin, and Yann Le Cun. "Measuring the VC-dimension of a learning machine". In: *Neural Computation* 6.5 (1994), pp. 851–876.

[193] Santosh Vempala. "Geometric random walks: a survey". In: *Combinatorial and Computational Geometry* 52.573-612 (2005), p. 2.

[194] William E Vinje and Jack L Gallant. "Natural stimulation of the nonclassical receptive field increases information transmission efficiency in V1". In: *Journal of Neuroscience* 22.7 (2002), pp. 2904–2915.

[195] Ziyu Wang, Shakir Mohamed, and Nando Freitas. "Adaptive Hamiltonian and Riemann manifold Monte Carlo". In: *International Conference on Machine Learning*. 2013, pp. 1462–1470.

[196] Ben D B Willmore, Ryan J Prenger, and Jack L Gallant. "Neural representation of natural images in visual area V2". In: *The Journal of neuroscience* 30.6 (2010), pp. 2102–2114.

[197] Ben DB Willmore, James A Mazer, and Jack L Gallant. "Sparse coding in striate and extrastriate visual cortex". In: *Journal of neurophysiology* 105.6 (2011), pp. 2907–2919.

[198] William H Wolberg and Olvi L Mangasarian. "Multisurface method of pattern separation for medical diagnosis applied to breast cytology." In: *Proceedings of the national academy of sciences* 87.23 (1990), pp. 9193–9196.

[199] Blake E Woodworth and Nati Srebro. "Tight complexity bounds for optimizing composite objectives". In: *Advances in neural information processing systems*. 2016, pp. 3639–3647.

[200] Changye Wu, Julien Stoehr, and Christian P Robert. "Faster Hamiltonian Monte Carlo by Learning Leapfrog Scale". In: *arXiv preprint arXiv:1810.04449* (2018).

[201] Michael C-K Wu, Stephen V David, and Jack L Gallant. "Complete functional characterization of sensory neurons by system identification". In: *Annu. Rev. Neurosci.* 29 (2006), pp. 477–505.

[202] Tatiana Xifara et al. "Langevin diffusions and the Metropolis-adjusted Langevin algorithm". In: *Statistics & Probability Letters* 91 (2014), pp. 14–19.

[203] Daniel L K Yamins and James J DiCarlo. "Using goal-driven deep learning models to understand sensory cortex". In: *Nature Neuroscience* 19.3 (2016), pp. 356–365.

[204] Daniel L K Yamins et al. "Performance-optimized hierarchical models predict neural responses in higher visual cortex". In: *Proceedings of the National Academy of Sciences* 111.23 (2014), pp. 8619–8624.

[205] Jason Yosinski et al. "How transferable are features in deep neural networks?" In: *Advances in neural information processing systems*. 2014, pp. 3320–3328.

[206] Jason Yosinski et al. "Understanding neural networks through deep visualization". In: *arXiv preprint arXiv:1506.06579* (2015).

[207] Bin Yu and Per Mykland. "Looking at Markov samplers through cusum path plots: a simple diagnostic idea". In: *Statistics and Computing* 8.3 (1998), pp. 275–286.

[208] Matthew D Zeiler and Rob Fergus. "Visualizing and understanding convolutional networks". In: *Computer vision–ECCV 2014*. Springer, 2014, pp. 818–833.

[209] Peng Zhao and Bin Yu. "On model selection consistency of LASSO". In: *The Journal of Machine Learning Research* 7 (2006), pp. 2541–2563.